



Reference Manual

Book I: Installation and Script

Version 9.0 for Microsoft Windows®

Third Edition, Lund, Sweden, February 2010

Authored by QlikTech International AB /HIC/KHN/JNN/MSJ/CEN

Copyright © 1994-2010 Qlik®Tech International AB, Sweden.

Under international copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written permission of QlikTech International AB, except in the manner described in the software agreement.

Qlik®Tech and Qlik®View are registered trademarks of QlikTech International AB.

Microsoft, MS-DOS, Windows, Windows NT, Windows 2000, Windows 2003, Windows XP, Windows Vista, Windows 2008, SQL Server, FoxPro, Excel, Access and MS Query are trademarks of Microsoft Corporation.

IBM, AS/400 and PowerPC are trademarks of International Business Machines Corporation.

Borland, Paradox and dBASE are trademarks of Borland International.

ORACLE and SQL*Net are trademarks of Oracle Corporation.

MacOS is a trademark of Apple Corporation.

Sun Java and JavaBeans are trademarks of Sun Microsystems, Inc.

OVERVIEW BOOK I

INTRODUCTION

I

INSTALLATION

II

GENERAL

III

SCRIPT

VI



CONTENT

PART I: INTRODUCTION

1	BEFORE YOU BEGIN	13
	1.1 About This Book	14
	1.2 Conventions	15
	1.3 QlikTech Support Services	16
2	WHAT'S NEW IN QLIKVIEW 9	17
	2.1 Licensing	17
	2.2 Server and document administration	18
	2.3 AccessPoint	19
	2.4 Collaboration	20
	2.5 Performance and scalability	20
	2.6 Real-time data update	21
	2.7 Integrated reporting	21
	2.8 Script features	21
	2.9 General features	22
	2.10 Layout features	23
	2.11 Chart features	24
	2.12 AJAX client capabilities	25
	2.13 Java client capabilities	27
	2.14 Mobile clients	27
3	COMPATIBILITY ISSUES FOR EARLIER VERSIONS OF QLIKVIEW	29
	3.1 File Format Compatibility	29
	3.2 Mixed Client/Server Environments	30
	3.3 Automation Change Patterns	30

PART II: INSTALLATION

4	INSTALLING QLIKVIEW	35
	4.1 Starting the Installation Program	35
	4.2 When the Installation Program Is Running	35
	4.3 Serialization	36

4.4 Network distribution of QlikView	37
4.5 License Enabler File Editor	37
4.6 Updating QlikView	38
5 OLE DB AND ODBC	39
5.1 OLE DB	39
5.2 When Is the ODBC Interface Needed?	39
5.3 Creating ODBC Data Sources	41

PART III: GENERAL

6 STARTING QLIKVIEW	45
6.1 Command Line Syntax	45
6.2 The Start Page	46
6.3 Connect to Server	49
6.4 Batch Execution	51
7 QLIKVIEW DOCUMENTS	55
7.1 File Compatibility	55
8 MENU COMMANDS	57
8.1 File Menu	57
8.2 Edit Menu	62
8.3 View Menu	64
8.4 Selections Menu	65
8.5 Layout Menu	66
8.6 Settings Menu	68
8.7 Bookmarks Menu	69
8.8 Reports Menu	70
8.9 Tools Menu	71
8.10 Object Menu	72
8.11 Window Menu	72
8.12 Help Menu	74
9 THE TOOLBARS AND THE STATUS BAR	77
9.1 The Toolbars	77
9.2 The Standard Toolbar	77
9.3 The Navigation Toolbar	80
9.4 The Design Toolbar	81
9.5 The Sheet Toolbar	84
9.6 The Bookmark Toolbar	84

9.7 Configuration of the Toolbars	84
9.8 The Status Bar	89
10 USER PREFERENCES	91
10.1 General	91
10.2 Save	97
10.3 Editor	99
10.4 Design	104
10.5 Objects	107
10.6 Font	110
10.7 Export	111
10.8 Printing	116
10.9 Mail	118
10.10 Locations	120
10.11 Security	122
10.12 License	124
11 CALCULATION MEMORY SETTINGS	125
12 LOGIC AND SELECTIONS	129
12.1 Single Field Value Selection	129
12.2 Multiple Selection within a Field	130
12.3 Search	130
12.4 The Advanced Search Dialog	133
12.5 Moving the Selection	135
12.6 Logical And Mode	137
12.7 Locking Selected Field Values	138
12.8 Selections in Other Objects	138
12.9 Current Selections	141
12.10 Selection Styles	142
12.11 Loosely Coupled Tables	144
13 BOOKMARKS	149
13.1 About bookmarks	149
13.2 Creating bookmarks	149
13.3 Applying bookmarks	152
13.4 Bookmarks dialog	153
13.5 Exporting Bookmark(s)	157
13.6 Importing Bookmark(s)	157
14 REPORTS	161
14.1 About Reports	161
14.2 The Reports Menu	162
14.3 The Report Editor Dialog	162

15 ALERTS	183
15.1 About alerts	183
15.2 Creating alerts	183
15.3 Using alerts	183
15.4 The Alerts dialog	185
15.5 The Alert Wizard	189
16 VARIABLE OVERVIEW DIALOG	199
17 EXPRESSION OVERVIEW DIALOG	201
18 INTERNAL FILES	205
18.1 File references in the QlikView layout	205
18.2 Internal files in QlikView documents	205
18.3 Internal files in the QlikView program file	205
18.4 Bundled Sounds	208

PART IV: SCRIPT

19 VARIABLES AND FIELDS	213
19.1 Fields	213
19.2 Input Fields	213
19.3 Variables	214
19.4 Other Entities in QlikView	215
20 SCRIPT DIALOG BOXES	217
20.1 Edit Script	217
20.2 Hidden Script	227
20.3 Table Viewer	228
20.4 Data Link Properties	231
20.5 Create Select Statement	233
20.6 Open QlikView File	238
20.7 Open Local Files	238
20.8 Open Internet Files	239
20.9 Inline Data Wizard	241
20.10 Access Restriction Table Wizard	244
20.11 File Wizard	245
21 SCRIPT SYNTAX	281
21.1 Statements and Keywords	281
21.2 Script Keywords and Statements in Alphabetical Order	281

21.3 Table Names	342
21.4 Script Variables	343
21.5 Dollar-Sign Expansions	350
21.6 Quotation Marks	353
22 SCRIPT EXPRESSIONS	355
22.1 Expression Syntax	355
22.2 Operators	356
22.3 Aggregation Functions	359
22.4 Other Functions	383
23 DATA STRUCTURES	461
23.1 Data Loading Statements	461
23.2 Execution of the Script	461
23.3 System Fields	461
23.4 Logical Tables	462
23.5 Associations between Logical Tables	463
23.6 Renaming Fields	466
23.7 Concatenating Several Tables into One	467
23.8 Join and Keep	469
24 EVALUATING THE LOADED DATA	475
24.1 Generic Databases	475
24.2 Cross Tables	476
24.3 Matching Intervals to Discrete Data	477
24.4 Using the Extended IntervalMatch Syntax to Resolve Slowly Changing Dimension Problems.	479
24.5 Hierarchies	481
24.6 Semantic Links	483
24.7 Linking Information to Field Values	487
24.8 Data Cleansing	488
24.9 Wild Cards in the Data	491
24.10 NULL Value Handling in QlikView	492
24.11 Character Sets	495
25 QVD FILES	497
25.1 About QVD Files	497
25.2 The QVD file Format	497
25.3 What are QVD files good for?	497
25.4 How Fast Is It?	498
25.5 Creation of QVD Files	498
25.6 Reading data from QVD files	500
25.7 Using QVD files for incremental load	501

26 SECURITY	505
26.1 Authentication and Authorization	505
26.2 Security using the QlikView Publisher	505
26.3 Security using the Section Access in the QlikView script	506
26.4 Sections in the Script	506
26.5 Access Levels in Section Access	506
26.6 Section Access system fields	507
26.7 Restrictions on QlikView functionality	509
26.8 Dynamic Data Reduction	510
26.9 Inherited Access Restrictions	511
26.10 Encryption	511

PART I: INTRODUCTION

- **About QlikView**
- **About This Book**
- **What's New in QlikView 9?**



1 BEFORE YOU BEGIN

Welcome to QlikView - the data access solution that enables you to analyze and use information from different data sources.

With QlikView, it is easy to grasp the overall picture and spot the connections, even when working with large and complex data sets. You can integrate information from different sources and the information can quickly be made available through the network. The right information gets to the right person. The associative technology allows you to create a unique interface for interactive presentation and analysis of any kind of information.

QlikView manages information in a way that resembles and supports the way the human brain works. Just like the human brain, it gradually makes associative connections in the information being processed. You - not the database - decide which questions to ask. Just click on the item you want to know more about.

Conventional information search systems often require a top-down approach, while QlikView allows you to get started with any piece of data regardless of its location in the data structure.

The retrieval of data in conventional systems is often a complex task requiring extensive knowledge of the structure of the databases and of the syntax of the query language. The user is frequently limited to predefined search routines. QlikView revolutionizes this by making it possible to select freely from data displayed on the screen with a click of the mouse.

QlikView has many areas of application. You are the one to decide how the program is to be used. QlikView helps you acquire a unified and coherent overview of the data in different databases - your own or someone else's, central or local. QlikView can be used with virtually any database.

With QlikView you can

- create a flexible end user interface to an information warehouse
- get snapshots of data relations
- make presentations based on your data
- create dynamic graphical charts and tables
- perform statistical analysis
- link descriptions and multimedia to your data
- build your own expert systems
- create new tables, merging information from several sources

-
- build your own business intelligence system

Some examples of QlikView applications being used today are financial systems, human resources administration, market analysis, customer support, project administration, production control, stock inventories and purchasing. You can even mix the different applications to gain entirely new information overviews.

1.1 About This Book

This Reference Manual consists of three books: *Installation and Script*; *Layout, Number Formats and Macros*; and *Charts*. Book I contains four parts: *Introduction, Installation, General* and *Script*. Book II consists of the parts *Sheets and Sheet Objects, Number Formats and Macros* and *Automation*. Book III consists of the part *Charts*, as well as of a *Glossary* and an *Appendix*.

In addition, we provide a separate *Tutorial*, guiding you step by step through the main features of QlikView.

The *Introduction* part, which you are currently reading, contains general information on the program and on our support services.

The *Installation* part shows how to install QlikView and other components needed to run the program.

The *General* part shows how to start QlikView, introduces the QlikView menu bar and toolbars, and describes how selections are made.

The *Script* part contains all the information you need to load data into QlikView. This part explains the dialog boxes involved in the creation of scripts, and provides a detailed syntax description. Access restriction is also handled here.

The interpretation and formatting of date, time and number formats is treated in a separate part, *Number Formats*. Here you will find explanations of the number format dialogs, as well as general information on how different formats are handled by QlikView.

The *Macros and Automation* part provides a brief description of the Automation interface, as well as of the use and creation of macros.

The *Sheets and Sheet objects* part provides information on how to create and modify the layout. This part ends with a chapter on exporting and printing sheet objects.

Since there is a great number of different chart types, these are treated in a separate part, *Charts*. Besides the variety of settings used for creating and modifying charts, this part also provides detailed information on chart expressions.

The *Glossary* explains some of the terminology used in the manual.

Finally, the *Appendix* contains chapters on issues such as data security, sharing QlikView documents in workgroups, etc.

A Help file is installed together with the program. You can open this file from many places in the program, and it will assist you in solving most of the problems encountered.

1.2 Conventions

Before you start using QlikView, it is important to understand the terms and notational conventions used in the documentation. In this section some of the terms will be explained.

General Conventions

- The word "choose" is used for carrying out a menu command in the toolbar or in a dialog.
- The word "select" is used for highlighting an object in a list or on a sheet that you want your next action to affect. It is also used for highlighting field values, thereby making logical selections within the data.
- Numbered lists (e.g. 1, 2, 3, ...) indicate procedures with two or more sequential steps.
- Bulleted lists, such as this one, provide information, and do not indicate procedural steps.

Mouse Conventions

- The descriptions in this manual assume that you have configured the left mouse button as the primary mouse button and the right mouse button as the secondary mouse button (This is default in Windows).
- "Point at..." means move the mouse and thus the cursor until the tip of the cursor points at the referred object.
- "Click..." means point at the referred object, then press and immediately release the mouse button without moving the mouse.
- "Double-click..." means click the mouse button twice in rapid succession.
- "Right-click" means click with the right mouse button.

Keyboard Conventions

- Key names appear in small capital letters, e.g. "Press ENTER".
- The RETURN key and the ENTER key perform the same action in QlikView.

-
- A plus sign "+" used between two key names indicates that you must press both keys at the same time. E.g., CTRL+S means that you should press the CTRL-key while pressing S.
 - A comma sign "," used between two key names indicates that you must press the keys sequentially.

1.3 QlikTech Support Services

Contact us if you need product support, additional training or consultation concerning application development. Please consult our homepage for current information on how to get in touch with our support services. You will find us at:

<http://www.qlikview.com>.

QlikTech International Headquarters

QlikTech International Phone: +1 (888)-828-9768
150 N. Radnor Chester Road Fax: 610-975-5987
Suite E220
Radnor, PA 19087
USA

For other locations please visit our www home page (see above).

2 WHAT'S NEW IN QLIKVIEW 9

The QlikView development team has again listened to the ideas and suggestions of users, customers and partners. As a result of our development and your comments, we are now ready to present QlikView 9 with more than one hundred new features and improvements.

We believe that there will be something for everyone to enjoy. Most of the new features are presented more in detail below.

2.1 Licensing

Test Server license

A special license type has been created for use with QlikView Server for test purposes. A QVS running with such a license will have the full feature set and performance, but the word "Test" will be superimposed on all charts and added to all object captions. See the QlikView Server Reference Manual for more information.

QVS migration mode

A setting has been added in the QVS Management Console allowing you to temporarily run two clusters of QVSs, sharing the same license during migration between versions. See the QlikView Server Reference Manual for more information.

QlikView Personal Edition

With QlikView 9.0, everyone is free to create QlikView documents for personal use. Hence, a normal QlikView desktop can be run without purchasing a license. There are no limitations, except that it is not possible to open QlikView documents created by other users. Nor is it possible to import an entire layout from an xml file.

Technically, every QlikView document is now saved with a user key - information tied to the user and the computer. A QlikView Server or a user of a full edition QlikView (a bought license) can open any file, but a user of QlikView Personal Edition can only open files with the correct user key.

Should you change computer, you will not be able to open your old documents with a QlikView Personal Edition. However, you will be asked to recover your files instead. The difference is that a new user key is generated and used for subsequent files. QlikView will accept four (4) user key changes.

Collapse of Developer, Professional and Analyzer

A consequence of the introduction of QlikView Personal Edition is that the difference between Developer, Professional and stand-alone Analyzer is no longer relevant and the three levels have been removed. Everyone has the right to run a QlikView with developer capability.

However, the functionality of corporate users can still be limited through deployment options stored in the registry.

Evaluation

The 15-day evaluation of QlikView Developer has been removed.

License lease from QlikView Server

The standard way to distribute a QlikView User license is to use a Named CAL on a server. The client will then get a license lease from the server, valid for 30 days and enabling all functionality of the QlikView Desktop.

2.2 Server and document administration

QlikView Server with integrated reload

QlikView Server now comes with integrated reload scheduling capabilities. The product previously known as Publisher Standard Edition has been discontinued, while the capabilities it offered are part of the standard QlikView Server install package. See the QlikView Server Reference Manual for more information.

QlikView Publisher improvements

The product previously known as QlikView Publisher Enterprise Edition is now an add-on license component to QlikView Server.

The Publisher module has been optimized for more efficient utilization of multiple execution instances.

QlikView Management Console (QMC)

The new integrated QlikView Management Console is the result of customer feedback and major usability studies. Completely built around modern AJAX technology, it will run in any browser and without reliance on e.g. IIS. We have strived to reduce the number of available settings to those most commonly used by customers, thereby producing a cleaner, more intuitive interface.

Even without the Publisher module, the QMC will feature a page for basic reload scheduling. If the Publisher module is installed this will be expanded to a wizard style interface for setting up more advanced Publisher tasks.

The QMC handles only one instance of QlikView Server and one execution instance of the Publisher. See the QlikView Server Reference Manual for more information.

QlikView Enterprise Management Console (QEMC)

The QEMC gives you full access to all possible settings for QlikView Server and the Publisher module. It also lets you control multiple instances of QlikView Server and multiple Publisher execution instances from a single management console, by means of an integrated tree-control.

Just as the QMC the QEMC is built around AJAX technology and will run in any browser. Also here extensive usability studies have been done prior to implementation. See the QlikView Server Reference Manual for more information.

Document default scheduling for reload

When designing a QlikView document it is now possible to define a default reload schedule in the **Document Properties** dialog (see page 37 in Book II). This can in many cases reduce the administrative overhead of setting up a reload task when adding new documents to the server. From the QMC or QEMC you may accept a proposed schedule with a single click.

QlikView Server SNMP support

QlikView Server is now enabled for monitoring from third-party software via the SNMP protocol. See the QlikView Server Reference Manual for more information.

2.3 AccessPoint

New QlikView AccessPoint

The all new QlikView AccessPoint has been designed as the principal interface where all users should find and open their QlikView documents.

QlikView 9 has the ability to bundle essential document metadata including active sheet thumbnail (optional) so that QlikView AccessPoint can use it for presentation of the documents. The resulting GUI is less technical than its QlikView 8 counterpart and more usable.

Improved load balancing support

QlikView AccessPoint now has the ability to sense current load on several connected QlikView Servers and direct the user to the most appropriate server instance.

2.4 Collaboration

Full collaboration support in thin clients

The possibilities to create and share user sheet objects have been extended to the AJAX and Java clients. The new functionality includes the possibility to access sheet object properties.

User sheets

In addition to user sheet objects there is now also the possibility to create user sheets. Read more on page 66.

Granular sharing

Previously, sharing a user object always meant making it available to all other users of the document on a server. With QlikView 9 it also becomes possible to share to a selected list of users. Read more on page 255 in Book II.

Collaboration object undo/redo

The layout undo/redo function has been extended to user objects.

Updated collaboration pane

The collaboration pane has been updated to cater for the new possibilities listed above. Read more on page 255 in Book II.

2.5 Performance and scalability

Removal of 2 billion records limit

The hard limit of maximum 2 billion records in a table has been removed. The amount of data that can be handled in QlikView 9 is only limited by available RAM and CPU power.

Selection and calculation performance

Major optimizations have been made in the areas of field frequency calculations, show alternatives and for common chart aggregation operations.

RAM utilization performance

Storage and caching algorithms have been optimized to use available RAM more efficiently in QlikView Server.

2.6 Real-time data update

Dynamic Data Update

For the first time in QlikView history it is possible to programmatically update field data in real-time without running the script. Via new APIs and more or less standard SQL insert, update and delete commands, any QlikView field data can be updated directly in RAM. On a server the updated data is pushed out to the clients.

This functionality is freely available in stand-alone QlikView but requires an additional license on QlikView Server. See the QlikView API Guide for more information.

2.7 Integrated reporting

Report distribution from QlikView Publisher

QlikView Publisher can now be used for setting up tasks mailing PDF documents with predefined QlikView reports.

Report printing from AJAX and Java clients

QlikView reports can now be printed from both the java and the AJAX clients.

2.8 Script features

New table file wizard

The table file wizard has been completely revised for improved usability and function. See page 245 for more information.

Improved select statement wizard

The select statement wizard now supports the generation of multiple select statements. See page 233 for more information.

Load from field

The load statement may now use a previously loaded QlikView field as data source. This feature can be quite useful e.g. when reading data from web pages. Read more on page 309 and on page 246.

New script functions

The new **Attribute** function allows you to extract file attributes, such as id3-tags from mp3 files and exif tags from jpg files. The function currently supports attribute extraction from mp3, wma, wmv and jpg files. See page 421.

The new **GetFolderPath** function can be used to extract the path to a wide range of Windows standard folders, e.g. My Documents etc. See page 423.

Store into text file

The store into statement has been amended to allow output to text files with a delimiter of your choice. Read more on page 336 for more information.

New option for log files

An option has been added allowing you to retain multiple versions of script log files, each named with a time stamp. Read more on page 19 in Book II.

2.9 General features

Clear button

The function of the **Clear** button has been improved. As default, the **Clear** button will work just the way it always has. However, you may at any time set any selection state as start state and use this as **Clear**. On a server deployment it is also possible for client users to set their own override **Clear State**. See page 80 for more information.

Export/import documents as xml

The possibility to export and import qvw documents, sheets and reports (excluding data) as xml has been added. This feature may e.g. be used for facilitating use of third party version control software with qvw files, eliminating the need to check in very large data sets. See page 61 and page 167 for more information.

Preview in table viewer

It is now possible to see a sample of table data by right-clicking on a table in the table viewer. Read more on page 230.

Relative change in input fields

The input syntax in input field cells has been extended, so that you can easily express relative changes and changes in percents. Read more on page 213.

Document chaining

A new action (see page 201 in Book II) has been added making it possible to open a new document from inside another document and apply current selections from the old document to the new document. This feature of course works both in stand-alone operation as when working from a QlikView Server.

2.10 Layout features

Actions

Action is a new entity in QlikView 9. They are derived from the old button shortcuts, which they also replace. Apart from offering a much wider range of operations than the old shortcuts (including most common operations on sheets, sheet objects, fields and variables), you may also define a series of operations within a single action.

The introduction of actions should greatly reduce the need for macros, which is good since macros are never efficient from a performance point-of-view. See page 198 for more information in Book II.

Gauge, text object and line/arrow actions

The new actions can not only be used on buttons. Also text objects, line/arrow objects and gauge charts can be given actions, which are executed when clicking on the sheet object in question.

Trigger actions

The trigger macros of previous versions of QlikView have been replaced by trigger actions (see under actions above). This gives you the possibility to build quite elaborate triggers without the use of macros.

Trigger macros from previous versions will be automatically translated to a RunMacro action when loaded into QlikView 9. Read more about Triggers on page 45 and page 87 in Book II.

Search Object

With the new search object you can search simultaneously in multiple fields (or even all fields in the document). Results are dynamically presented, as

you type, in a tree-view list field by field. Selections can then be made in the usual manner. See page 241 in Book II.

Search Substring Highlight

The search result from a text search in list boxes or search objects are now displayed with the match of the search substring highlighted in the result list.

Tree-view list box

The logical extension of the hierarchy resolution script syntax introduced in QlikView 8.5. When displaying a field containing hierarchic data in a list box set to this mode, you get a full tree-view structure, where you can expand, collapse and make selections in either nodes or leaves. See page 102 in Book II.

2.11 Chart features

Indirect set analysis

Set analysis in QlikView 8.5, although very powerful, was restricted to stating direct selections in a field. With the new extensions in QlikView 9 it also becomes possible to state selections in a field based on selections in another field. You may e.g. express selections such as “Select all possible values in Customers based on a Sales last year”. See page 353 in Book III for more information.

Mini Charts

Mini charts are small line/bar charts inside a table cell. You specify an extra dimension, separate from the main chart dimensions, over which the expression is aggregated and plotted as a miniature chart. Read more on page 238 in Book III.

Trellis charts

Trellis chart means using the first dimension of a chart as iteration dimension for producing an array of charts. Any type of bitmap chart can be made into a trellis display. Read more on page 38 in Book III.

Dynamic chart backgrounds

The background of a chart can now be dynamically calculated. This opens up possibilities e.g. for map backgrounds that update with selections. Read more on page 70 in Book III.

Show condition on straight table columns

Instead of just having fixed hide or show status, you may now specify a dynamically calculated show condition for straight table columns. Read more on page 243 in Book III.

Clickable URL links in table cells

A new representation mode 'link' lets you transform expression cells in straight and pivot tables into clickable links. The normal selection on click behavior is disabled in such cells. It is possible to specify a display text and a URL for the link in the chart expression. The display text will appear underlined, to indicate link status. Read more on page 215 in Book III.

Box plot wizard

A new wizard has been added to facilitate the creation of box plot charts. Learn more about the wizard on page 261 in Book III.

Upgraded defaults and themes

QlikView 9 includes several improvements in terms of defaults for new objects and upgraded design themes.

2.12 AJAX client capabilities

Pre-generation of AJAX pages no longer needed

The need to pre-generate AJAX pages for QlikView Server has disappeared. QlikView Server will display documents in AJAX when called upon, just like it has always done with the Java and plug-in clients. This takes away most of the migration worries between versions, which have so far been a potential problem for QlikView AJAX deployments.

Collaboration objects

The AJAX client now has full support for all types of user objects and shared objects. The new user sheets are also supported.

Property dialogs

Property dialogs are available for modification of user objects. Instead of just copying the property dialogs from the installed clients, we have designed a new type of dialogs, which we believe will be easier to use. It is for example possible to have property dialogs open for several objects at the same time and e.g. drag and drop expressions between them.

Layout compliance with installed clients

Additional layout features previously not supported in the AJAX client have been added, e.g. support for

- OnOpen message alerts,
- gauges in table cells,
- images in list boxes,
- semantic list boxes,
- x-axis scroll in charts
- balloon pop-ups in charts,
- right-click menu on cycle icons,
- right-click menu on fast type change icons,
- free-floating current selections window,
- bookmark toolbar,
- maximization of sheet objects,
- automatic truncation of text in cells (as in installed client).

Many graphical improvements have been made, e.g. a complete overhaul of the multi box.

New layout features

In addition to the above new layout features from QV9 have naturally been implemented also in the AJAX client. This includes e.g. the new Clear button, mini charts in tables, button actions, tree-view list box, clickable URL links in table charts and the new search object.

Report printing

QlikView reports can be printed from the AJAX client.

Performance improvements

The AJAX client has received further optimizations for considerably improved performance in terms of speed and rendering.

2.13 Java client capabilities

Collaboration objects

The Java client now has full support for user objects and shared objects. Property dialogs are available for modification of objects as well as undo/redo commands and design grid. The new user sheets are also supported.

New layout features

Additional layout features previously not supported in the Java client have been added, e.g. support for gauges in table cells images in list boxes.

In addition to this new layout features from QV9 have naturally been implemented also in the Java client. This includes e.g. the new Clear button, mini charts in tables, button actions, tree-view list box and clickable URL links in table charts.

Report printing

QlikView reports can be printed from the Java client.

Performance improvements

The Java client has received further optimizations for improved performance in both communication and rendering.

The UI look has been consolidated to look the same regardless of which platform the client is run under.

2.14 Mobile clients

For the first time it will be possible to bring along the power of QlikView analysis on your mobile phone. Our approach is threefold, in order to cover as many platforms and needs as possible.

Iphone client

The native Iphone client available from Apple's AppStore offers a completely new interaction paradigm for QlikView. It optimizes the use of the limited screen real estate while maximizing the use of the Iphone's innovative user interface. Sheets are no longer graphical entities, but rather used as a collection of sheet objects. You navigate between sheet objects using cover-flow, having one object in focus at any time but still getting context and feedback via miniatures of the other objects below. Turn the Iphone to horizontal position to zoom in on an object, shake to clear selections and so on...

Java Mobile client

Whereas the Iphone client will work only on Iphone (and Ipod Touch), the Java Mobile (J2ME) client will work on the vast majority of all current mobile (smart)phones, including Blackberry and Symbian phones. The main requirement is that the phone has a Java Virtual Machine. This client is more conventional and presents the QlikView sheets more or less as you know them from our traditional clients. Although the experience will be best on large touch-screens, the client can actually be used even on phones with only a numeric keypad.

AJAX on mobiles

Finally there is the option of running our standard AJAX client on mobile devices. We see a clear trend that more and more mobile phone browsers will handle modern AJAX applications in an acceptable manner. Mobile Safari on the Iphone was the first example, but others have followed. This solution may be considered on selected mobile platforms.

3 COMPATIBILITY ISSUES FOR EARLIER VERSIONS OF QLIKVIEW

The development team has done absolutely everything we could to make the migration to QlikView 9 from earlier versions as smooth as possible. We believe that we have succeeded and think that the migration from QlikView 7.52 and QlikView 8 has the potential of being very smooth indeed. This document lists some areas of concern.

3.1 File Format Compatibility

File Format Compatibility in QlikView 7, 8 and 9

QlikView 7.52 and later share the same file format as QlikView 9. You may work in these versions in parallel virtually without thinking about it. New QlikView 9 document features will of course not work in QlikView 7.52 or QlikView 8.x but will be retained even if the document is opened and edited in the earlier version. The whole thing works very much like Microsoft Office files have been working for some time now.

File Format Read Compatibility with Earlier Versions

QlikView 9 will open all QlikView files created in QlikView 7.52 and later. Should you need to open files from earlier QlikView versions, you need to go via QlikView 7.52. Contact your QlikView vendor for a free copy.

The development team always strives to make the appearance of a document in the new version as similar as is ever possible to how it looked in the older version. However, sometimes added functionality must come with the price of changes in the look of old documents.

Saving in Earlier File Formats from QlikView 9

QlikView 9 cannot save documents in any file format usable in versions prior to QlikView 7.52.

3.2 Mixed Client/Server Environments

QlikView 9<-> QlikView 7/8

QlikView Server 7.52 and later will work seamlessly with QlikView 9 clients as will QlikView Server 9 with QlikView 7.52 and later clients. Of course, in order to make use of QlikView 9 specific functionality you need to have both server and client of version 9.

QlikView 9 is not compatible with QlikView versions 7.51 and older.

Earlier Versions

QlikView Server 9 is not compatible with QlikView 6 clients. Nor are QlikView Server 6 compatible with QlikView clients.

3.3 Automation Change Patterns

This section contains patterns for changes necessary when using Automation code written for QlikView 6 in QlikView 9.

GraphProperties.ExpressionDatas

All references to the Automation API `GraphProperties.ExpressionDatas` must be changed according to the pattern shown in the example below:

QV6 code

```
set chart =
ActiveDocument.Sheets("Main").CreateComboChart
chart.AddDimension "ProductType"
chart.AddExpression "sum(Amount)"
chart.AddExpression "count(Customer)"
set p = chart.GetProperties
p.ExpressionDatas.Item(1).Accumulate = 1
chart.SetProperties p
```

QV9 code

```
set chart =
ActiveDocument.Sheets("Main").CreateComboChart
chart.AddDimension "ProductType"
chart.AddExpression "sum(Amount)"
chart.AddExpression "count(Customer)"
set p = chart.GetProperties
p.Expressions.Item(1).Item(0).Data.ExpressionData.Accumulate = 1
chart.SetProperties p
```

GraphProperties.ExpressionVisuals

All references to the Automation API GraphProperties.ExpressionVisuals must be changed according to the pattern shown in the example below:

QV6 code

```
set chart =
ActiveDocument.ActiveSheet.CreateComboChart
chart.AddDimension "ProductType"
chart.AddExpression "count(Customer)"
set p = chart.GetProperties
set expr = p.ExpressionVisuals.Item(0)
expr.ShowAsBar = false
expr.ShowAsLine = true
expr.ShowAsSymbol = true
chart.SetProperties p
```

QV9 code

```
set chart =
ActiveDocument.ActiveSheet.CreateComboChart
chart.AddDimension "ProductType"
chart.AddExpression "count(Customer)"
set p = chart.GetProperties
set expr =
p.Expressions.Item(0).Item(0).Data.ExpressionVisual
expr.ShowAsBar = false
```

ChartProperties.DataAxes

All references to the Automation API ChartProperties.DataAxes must be changed according to the pattern shown in the examples below:

QV6 code

```
set chart =
ActiveDocument.Sheets("Main").CreateLineChart
chart.AddDimension "ProductType"
chart.AddExpression "sum(Amount)"
set p = chart.GetProperties
p.ChartProperties.DataAxes(0).Logscale = true
chart.SetProperties p

set chart = ActiveDocument.GetSheetObject("CH01")
set p = chart.GetProperties
for i = 0 to p.ChartProperties.DataAxes.Count - 1
    p.ChartProperties.DataAxes.Item(i).Grid = true
next
chart.SetProperties p
```

QV9 code

```
set chart =
ActiveDocument.Sheets("Main").CreateLineChart
chart.AddDimension "ProductType"
chart.AddExpression "sum(Amount)"
set p = chart.GetProperties
p.Expressions.Item(0).Item(0).Data.AxisData.Logscale
= true
chart.SetProperties p

set chart = ActiveDocument.GetSheetObject("CH01")
set p = chart.GetProperties
for i = 0 to p.ChartProperties.DataAxes.Count - 1
    p.Expressions.Item(i).Item(0).Data.AxisData.Grid =
true
next
chart.SetProperties p
```


PART II: INSTALLATION

- Installing QlikView
- OLE DB and ODBC



4 INSTALLING QLIKVIEW

In your QlikView package you will find a DVD, a note with a serial number and a control number, and a license agreement. To install QlikView you will need the QlikView DVD, the serial number and the control number.

QlikView is designed to run on an IBM PC compatible computer with Windows 2003, Windows XP, Windows Vista or Windows 2008. For detailed information on the system requirements, see page 487 in Book III.

4.1 Starting the Installation Program

Do the following:

- 1 Insert the DVD into the DVD drive.
- 2 The setup program usually starts by itself: if it does not, open the Windows Explorer, find the DVD drive and double-click *Start.exe*.

The installation program should now have started.

4.2 When the Installation Program Is Running

- 1 In the first dialog that opens, choose your product type and click **OK**.
- 2 In the second dialog that opens, choose your language and click **OK**.
- 3 The third dialog that appears is just a “Welcome” dialog in the language that you selected in the previous dialog. Click **Next** to continue the installation.
- 4 You now get to the software license agreement. Read it, and click “I accept the license agreement” (if this is the case), then click **Next**.
- 5 In the **User Information** dialog you specify for whom the installation will be personalized. On Windows NT based operating systems, you must specify if the installation settings will be valid for the current account or if the settings are the same for any account that uses the computer (an account with administrator rights is required, if you choose the last alternative mentioned). Specify your choice and click **Next** to continue.

-
- 6 The installation program suggests a destination directory for the QlikView files. If you prefer to install to another folder, click **Browse** and choose another folder. To continue, click **Next**. If you install to the same folder as your previous version, the previous version will be overwritten by the installation.
 - 7 The installation type dialog appears. If you choose **Typical**, the following features will be installed: The QlikView program and its Help file, the Reference Manual (in PDF-format, Acrobat Reader is required to view it), the API Guide and Examples of Custom objects. If you choose **Complete**, the previous files mentioned will be installed plus the QlikView Tutorial and examples of QlikView functionality and features. If you choose **Custom**, a dialog where you can specify the features to install will appear when you click **Next**. Make your selections and click **Next**.
 - 8 If you selected **Typical** or **Complete** installation, you are now ready to start the installation, click **Next** to start. Otherwise you have to customize your installation before you are ready to install it.
 - 9 In the last dialog you click **Finish** to complete the installation process.

Note You always have to enter the QlikView license on the account that installed it; otherwise it might not function properly.

4.3 Serialization

If you get your QlikView license from a QlikView Server or if you have a Personal Edition of QlikView no serialization is necessary. It is however possible to acquire a serial number for this product and enter it in **User Preferences, License**. A registered QlikView will allow you to work offline without any contact with a Qlikview Server for longer periods (periods longer than 30 days). Contact your QlikView vendor for details.

Note The serialization process requires network access or optionally phone activation!

Once you have installed and ,optionally, serialized your copy of QlikView, you can start using the program.

4.4 Network distribution of QlikView

This type of installation is useful if you have many licenses and want to make sure that all users run the same version of QlikView. Each user still needs a separate license number.

To distribute a QlikView installation, you need the installation DVD. Serial numbers and control numbers are not necessary until you start your first QlikView session on the client computer.

QlikView 9 uses the Microsoft Windows Installer technique (MSI packaging) and a specific server installation program is no longer needed. The Microsoft Windows Installer technique has built-in support for this type of installations.

To find out more about the ways to install an MSI-package please visit the Microsoft homepage for Windows installer:

- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/windows_installer_start_page.asp

4.5 License Enabler File Editor

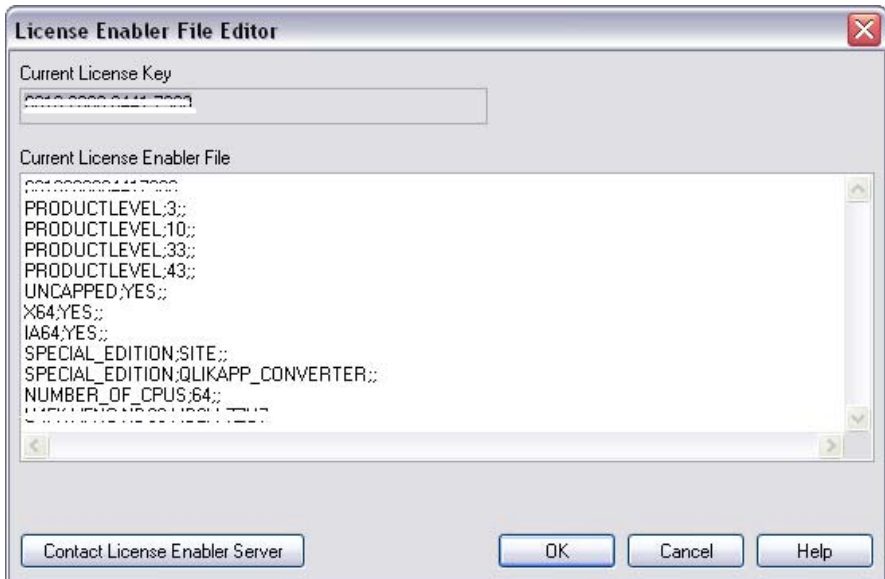


Figure 1. License Enabler File Editor

In this dialog the user can review or edit the License Enabler File.

Current License Key

The **Current License Key** of the QlikView license that is normally entered when the original program is installed. It may also be entered or edited at a later time via the **User Preferences, License** dialog. It is also possible to run Qlikview without a bought license key. In this case, you either have a license lease from a QlikView Server - a Named CAL - or you have a Personal Edition of QlikView.

Current License Enabler File

A text file containing information relevant to the installed license that is required in order to activate (enable) the license key.

Contact License Enabler Server

Click this button if the text edit box **Current License Enabler File** above is empty in order to contact the **License Enabler Server**.

Troubleshooting the License Enabler File

A License Enabler File (LEF file) is required in order to verify the validity of a QlikView license key. During the initialization of the license key the Qlik-Tech LEF Server is contacted via the Internet. Provided that the license information given checks out, a LEF file is automatically transferred to your computer. Under normal circumstances the LEF procedure is barely noticeable as it is done in the background. There are however instances when the procedure fails, either because you are unable to contact the LEF server or because a firewall prevents the transfer of the LEF file. When this happens the **License Failure** dialog will be displayed to inform you about the problem.

Should you be unable to obtain a valid LEF file through the normal procedure and you feel that you are entitled to one, you might instead obtain it directly from QlikTech Support. When you copy the text file directly into the LEF editor you must make sure that there are no trailing empty spaces in the text.

4.6 Updating QlikView

In the **Help** menu in QlikView, you find the option **QlikView Update...** Choosing this will take you to a QlikView update site on the Internet where you will be presented with possible updates for your QlikView version. What updates are available are based on information from your QlikView license and your operating system.

5 OLE DB AND ODBC

5.1 OLE DB

QlikView supports the OLE DB interface for connection to external data sources. You can access a great number of external databases via the Microsoft OLE DB Provider for ODBC Drivers.

5.2 When Is the ODBC Interface Needed?

It is necessary to have the ODBC (Open DataBase Connectivity) interface installed if you want to access a general database. The alternative is to export data from the database into a file that is readable to QlikView.

Normally some ODBC drivers are installed with the operating system. Additional drivers can be bought from software retailers, found on the Internet or delivered from your DBMS manufacturer. Some drivers are redistributed freely.

The ODBC interface described here is the interface on the client computer. If you plan to use ODBC to access a multi-user relational database on a network server, you might need additional DBMS software that allows a client to access the database on the server. Contact your DBMS supplier for more information on the software needed.

QlikView works with both 32-bit and 64-bit ODBC drivers. It is however very important to use the corresponding versions of the ODBC drivers and the program version, i.e. the 32-bit version of QlikView will only work with the 32-bit ODBC drivers and 64-bit QlikView will only work with the 64-bit ODBC drivers.

Is the Correct ODBC Driver Installed?

On a 64-bit platform it is possible to use both 32-bit and 64-bit applications. It is also possible to run both 32-bit and 64-bit ODBC drivers.

When using the 64-bit versions of ODBC and QlikView, the **ODBC Data Source Administrator** is found in the **Control Panel, Administrative Tools**.

When using the 32-bit versions of ODBC and QlikView, you must start the 32-bit administrator via the executable file, `odbcad32.exe`, that is located in the `SysWOW64` folder, usually `c:\windows\SysWOW64`.

According to Microsoft, in the 64-bit versions of the operating systems, the `system32` folder, usually `c:\windows\system32`, contains only 64-bit files. If

there is also a 32-bit version of the file, it resides in the syswow64 folder. See <http://technet.microsoft.com> for more information.

On a 32-bit operating system the configuration is very straightforward since all files and drivers are 32-bit.

There should be an **ODBC Data Source Administrator** installed on your system.(Classic view of the **Control Panel** is assumed):

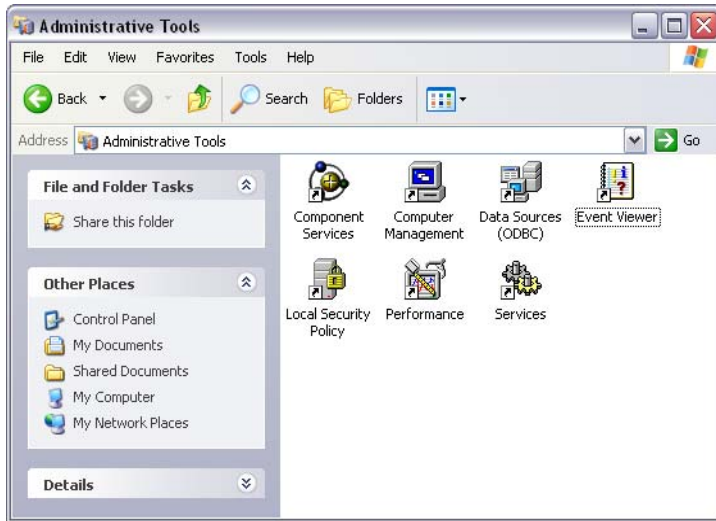


Figure 2. The Administrative Tools in the Windows XP Control Panel

- 1 Click the **Start** button on the task bar, choose **Control Panel**, then **Administrative Tools**.
- 2 Double-click the **ODBC** icon to find out what data sources you have defined. (For 32-bit ODBC on a 64-bit operating system, start `C:\windows\SysWOW64\odbcad32.exe`).

- 3 Go to the **ODBC Drivers** tab to find out what drivers are installed.

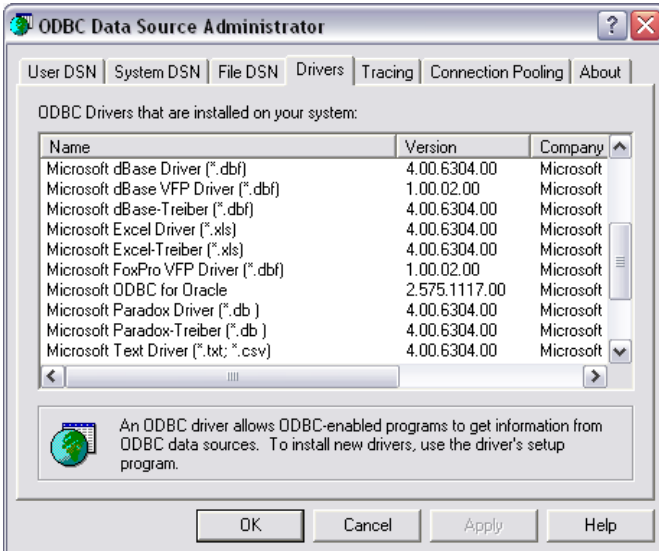


Figure 3. The ODBC Data Source Administrator dialog box, Drivers page

If the driver you need is not found on the ODBC Drivers dialog page, contact your software supplier to get the correct driver.

5.3 Creating ODBC Data Sources

You must create an ODBC data source for the database you wish to access. This can be done during the ODBC installation or at a later stage.

Before you start creating data sources, you must decide whether they should be user or system data sources. User data sources cannot be reached unless you have logged on with the correct user id. If you want to share your data sources with other users, you must create system data sources.

- 1 Open the **ODBC Data Source Administrator** dialog box again.
- 2 Go to the tab **User DSN** if you want to create user data sources, or to the tab **System DSN** if you want to create system data sources.
- 3 Click the **Add** button. The **Add Data Source** dialog box should appear, showing a list of the ODBC drivers installed.

-
- 4 If the correct ODBC driver is listed, select it and click the **OK** button. A dialog box specific to the selected database driver will appear. Name your data source and set the necessary parameters. Click the **OK** button when ready.

PART III: GENERAL

- **Menu Commands**
- **Toolbars and Status Bar**
- **User Preferences**
- **Logic and Selections**
- **Bookmarks**
- **Reports**
- **Alerts**



6 STARTING QLIKVIEW

Start QlikView from the **Start** menu, **Programs** or double-clicking on a QlikView document.



If you wish to start QlikView from the command line, you can use the parameters described in the following section.

6.1 Command Line Syntax

The QlikView command line invocation can be described by the following syntax (for a description of the syntax notation used, see page 523 in Book III):

```
[ path ] Qv.exe [ { switch } documentfile ]
```

The *path* is the path to the file, either absolute, or relative to the current directory.

documentfile::= [path] documentfilename

documentfilename is the name of your document file.

switch is used to mark different options:

/r

Reload switch. The document will be opened and the script reloaded. Thereafter the document will be saved, and QlikView will close itself.

/rp

The same as above, but with a partial reload.

/l

Load new data. The document will be opened and the script reloaded. The document will not be saved and QlikView will remain open.

/lp

The same as above, but with a partial reload.

/v

If this switch is immediately followed by a variable name and an assignment, the variable will obtain the assigned value before the script execution starts.

/nodata

Opens the document without variables, table and field data. This feature is also available for previously opened documents on the Start Page, see page 47.

/NoSecurity

Overrides QlikView security measures against hostile macros and scripts embedded in QlikView documents. No warning dialogs prompting the user to accept potentially harmful code will be shown. Use this switch with caution and only with well-known documents.

Example:

```
qv.exe /r /vMyvar=123 abc.qvw
```

Note In order to avoid error messages to appear when executing scripts in batch mode, always set the script variable `errormode` to 0 in the script (see page 349)

6.2 The Start Page

Once you have started QlikView according to the previous section, you will see the QlikView start page.

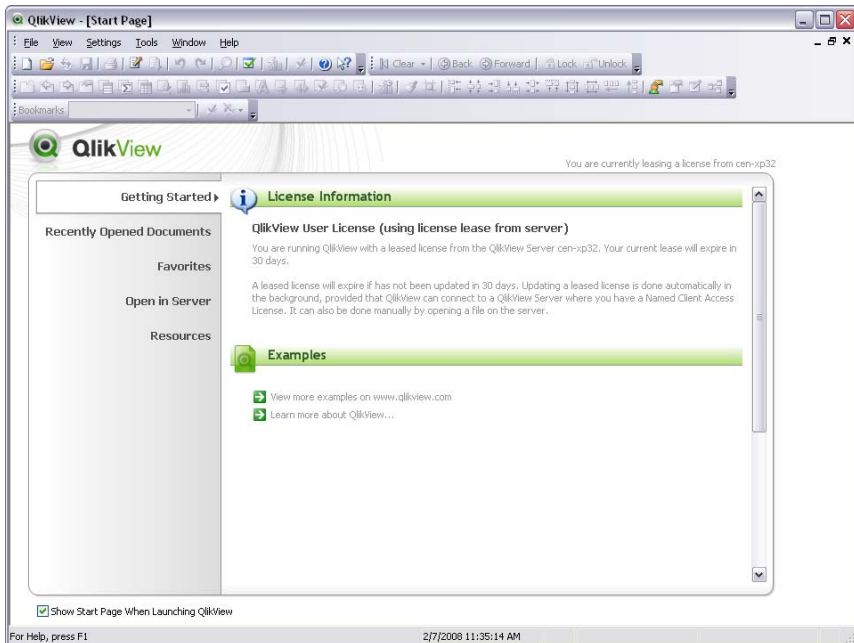


Figure 4. The Getting Started page of the QlikView start page

The QlikView start page contains several tabs. The number of pages and the contents on some of them may vary over time as they contain web content. The following pages will exist at all times:

The **Getting Started** tab contains information and links that can help you explore the wealth of possibilities in QlikView. This includes direct links to selected demo examples, a link to the QlikView demo example folder on your local computer and links to selected resources at www.qlikview.com.

The **Recently Opened Documents** tab contains a list of recently opened documents and web pages. Just click on one of them to re-open. By right-clicking on a document you gain access to a menu with the following useful commands:

Open "document" Without Data

Opens the document but skips variables, table and field data. What you get is a layout with all sheets and sheet objects in place but empty. This feature can be useful e.g. for opening corrupted documents or for avoiding long waits when opening very large documents for a small change in the layout (of course you need to re-run the script to re-populate the document after the changes). This can be done through the command line switch **nodata** as well, see page 45.

Open "document" and Reload Data

Opens the document and performs an immediate reload.

Browse Documents in Folder

Opens the folder containing the document in the list.

Add "Document" to Favorites

Adds the document or web page to your Favorites list.

Remove "Document" From This List

Removes the document from the list of recently used documents. The actual document file remains unchanged where it resides.

The **Favorites** tab contains a list of the documents and web pages previously added to your favorites list. Unlike the list of recently used documents the favorites list remains unchanged until you yourself explicitly add or delete entries. Just click on a favorite document or page to re-open. By right-clicking on a document you gain access to a menu with the following useful commands:

Open "Document" Without Data

Opens the document but skips table and field data. What you get is a layout with all sheets and sheet objects in place but empty. This feature can be useful e.g. for opening corrupted documents or for avoiding long waits when opening very large documents for a small

change in the layout (of course you need to re-run the script to re-populate the document after the changes).

Open "Document" and Reload Data

Opens the document and performs an immediate reload.

Browse Documents in Folder

Opens the folder containing the document in the list.

Remove "Document" From Favorites

removes the document from the Favorites list. The actual document file remains unchanged where it resides.

The **Open in Server** tab contains a list of recently used QlikView Servers and an easy-to-use interface for connecting to any other QlikView Server. Once you have selected a QlikView Server, QlikView will connect to it and show a list of available documents, which can then be opened with a single click. You may click on **Show Options >** for advanced authentication options. See the documentation of the **Connect to Server** dialog (below) for details on these commands.

Recent Servers shows a list of the servers you recently connected to.

Nearby Servers lets you see a list of all the servers on the network.

QlikView will remember which tab you last used and open the start page on that tab next time you start QlikView. If you don't want the start page to appear when you start the program, deselect the **Show Start Page When Launching QlikView** check box.

The start page can be left open and will then be found as **Start Page** on the **Windows** menu. If you close the start page, it can be re-opened at any time by choosing **Show Start Page** from the **Help** menu.

6.3 Connect to Server

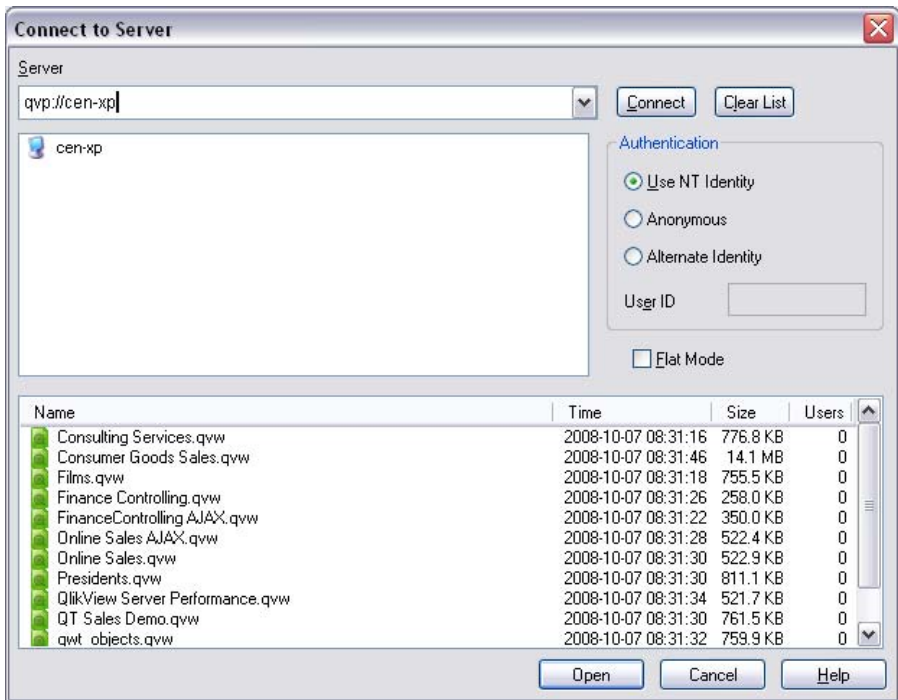


Figure 5. The Connect to Server dialog

This dialog is used for opening a QlikView document remotely on a QlikView Server. Note that when opening documents as a client to QlikView Server you will be restricted in what you can do with the document as compared to when opening local documents. As a client you may not be able to add or delete sheet objects, depending on whether the document supports collaboration. Sheets cannot be added or deleted. Furthermore you cannot access most properties dialogs, change macros, access the script or save the document.

Server

Enter the name or the URL to the computer running QlikView Server. Then click on **Connect** to connect. Recently visited servers can be retrieved in the drop-down list. A URL may contain additional connection data according to the syntax described in the next section.

Connect

Press this button in order to connect to a QlikView Server.

When connected, the central pane of the dialog will display the document folders available to you on the QlikView Server. Click on a folder and the available documents will be shown in the dialog's bottom pane. Double-click a document or select it and click on Open in order to open the document.

Clear List

This button clears the list of Recent Servers on the Start Page.

Authentication

Here you may select the way the QlikView Server will identify you when connecting. Three modes are available:

Use NT Identity

Use authenticated NT identity. Only works if you are working on a computer connected to the same Windows NT domain as the QlikView Server.

Anonymous

Use anonymous login. You will only be able to connect if the QlikView Server is set to allow anonymous users and you will only be able to see documents with file permissions for the QlikView Server anonymous account. The connection pseudo-URL under **Server** will be completed by a @ before the server name.

Alternate Identity

Use a specific NT user name known on the QlikView Server's domain. The user name is typed in the User ID edit box. You will be prompted for the corresponding user password when pressing the **Connect** button. The connection pseudo-URL under **Server** will be completed by *username*@ before the server name, where *username* is the name you use.

Flat mode

Mark this check box if you prefer the bottom pane of the dialog to display the contents of all subfolders simultaneously rather than using the tree-view approach.

Open

Opens the document selected in the bottom pane. Double-clicking the document name will have the same effect.

Cancel

Closes the dialog without opening a document.

Help

Opens interactive help.

Connection Pseudo URLs

When connecting to QlikView Server from Windows clients, either via the **Open in Server** dialog or via link files, a pseudo URL is used as document address.

The syntax is:

qvp://[[*username*][@]][*hostname*[:*port*];*protocol*]/*docname*

where

username is a Windows user ID. The parameter is optional.

hostname is the name of the host. The parameter is obligatory.

documentname is the name of the QlikView document (including the extension). The parameter is optional.

port (e.g. 4749) can be used to specify a specific port used by the server. The parameter is optional.

protocol (e.g. http) can be used to specify tunneling protocol. The parameter is optional.

@ without *username* denotes anonymous identity.

If user identity is omitted altogether, the logged in Windows identity is assumed.

6.4 Batch Execution

If the QlikView document should be updated regularly, e.g. every night, this is best done with QlikView Server or QlikView Publisher.

QlikView Server

QlikView Server provides a platform for hosting, and sharing QlikView information over the Internet/Intranet. QlikView Server is tightly integrated with QlikView to deliver a seamless suite of data analysis technology to end users. The server component of QlikView Server is the centerpiece of this technology, supplying a robust, centrally managed, QlikView document community, connecting multiple users, client types, documents, and objects within a secure and safe environment.

QlikView Publisher

QlikView Publisher is a member of the QlikView product family that manages content and access. By presenting your end-users with up-to-date information and letting you manage your QlikView documents in a powerful way, QlikView Publisher is a valuable addition to the QlikView suite.

QlikView Publisher distributes data stored in QlikView documents to users within and outside the organization. By reducing data, each user can be presented with the information that concerns him/her. The QlikView Publisher service and user interface are now fully integrated into QlikView Server and the QlikView Management Console (QMC).

If you do not have access to QlikView Server or QlikView Publisher, you can reload QlikView documents on the stand-alone application. Then you have to use the command line switches described in section 6.1.

Scheduling Programs

Windows has a **Task Scheduler** service. The services are controlled from the Computer management applet in the Control Panel (Windows Vista) or directly from the Control Panel (Windows XP). Make sure that **Status** is set to **Started** (and **Startup** to **Automatic** for the service to automatically start after a reboot) for the relevant service.

The easiest way to submit a batch job is usually to create a batch file with the relevant commands. The file, named e.g. *MyBatch.cmd*, could e.g. contain the following commands:

```
C:\>qlikview\qv.exe /r C:\qlikview\example\file.qvw
```

The Schedule Service

The schedule service normally runs under the system account and thus has some limitations. If you need to access network resources, reconfigure the service to run under a different account. Since the QlikView license is stored in the account-specific part of the registry, you will have to make sure that it is correctly registered for the account to be used. To do this you may submit an interactive job:

```
C:\>AT 11:00 /interactive "C:\Program  
Files\QlikView\qv.exe"
```

This will cause the **License Registration** dialog to appear. Here you can register the license also for the correct account.

The batch file (see the previous page) can be submitted in e.g. the following way:

```
C:\>AT 11:00 /every:m,t,w,th,f,sa,su  
C:\MyBatch.cmd
```

This will cause the batch file to be executed every day at 11 AM.

Section Access

Serial	User	Password	Access
CREATOR			Admin
4600 9999 9999 9999	Joe	ppp789	Admin
*	Joe	qqq456	User
*	User	rrr123	User

To reexecute a QlikView file containing a **section access**, make sure that the license serial number that is installed on the machine doing the batch execution is allowed to run the script without entering a user name or a password. See the example above. For more information about section access see “security” on page 505.

ODBC

If ODBC is used, please make sure that the DSN definition is available from the account doing the reexecution. The easiest way to assure this is probably to define the DSN as a System DSN in the ODBC Control Panel.



7 QLIKVIEW DOCUMENTS

A QlikView document is a file containing everything you need to analyze the data:

- the data itself
- the script needed to update the QlikView file with new data from the data source
- layout information, including all the sheets, list boxes, charts, etc.
- document alerts, document bookmarks and document reports
- access restriction information
- macro module

It is thus very easy to distribute information using QlikView files. The analysis can be made independently of the location of the original data or network conditions. QlikView documents are a way to share information with users that do not have access to the original data source.

7.1 File Compatibility

QlikView 7.52, 8 and 9 share the same file format. This means that documents can be opened and saved by the three versions without the need to bother about file formats.

Documents created with QlikView 7.52 and later can be opened with QlikView 9. In order to open files from earlier versions you must first install QlikView 7.52, open the document and save it in the 7.52 format.



8 MENU COMMANDS

The menus described in this chapter are found in the menu bar at the top of the screen. Most commands can also be configured as buttons in the toolbars, see “Configuration of the Toolbars” on page 84 for more information.

8.1 File Menu

The **File** menu contains the following commands :

New

Opens a new, empty document. This command can also be invoked via the keyboard shortcut CTRL+N.



Open...

Opens an existing file, either a QlikView file or a text file. If you open a text file, you will automatically get to the file wizard, which analyzes its contents and helps you to create a script. This command can also be invoked via the keyboard shortcut CTRL+O.



Open in Server...

Opens the **Connect to Server** dialog (see page 49). In this dialog you may connect to a QlikView Server and browse for a document to open in client mode. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+O.



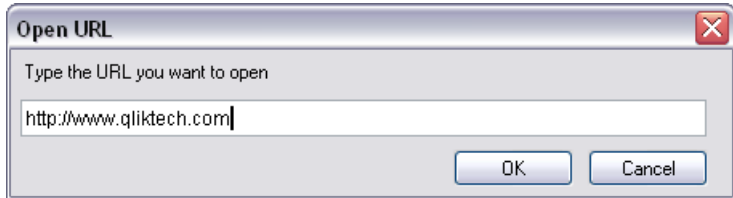
Note When opening documents as a client to QlikView Server you will be restricted in what you can do with the document compared to when opening local documents. As a client you may not be able to add or delete sheets and sheet objects, depending on whether or not the document supports collaboration. Furthermore, not all properties dialogs can be accessed, macros changed, the script accessed, data reloaded or the document saved.

Refresh Document

This command is only available with documents opened on QlikView Server and when there is a newer version of the document available on the server. When invoking a refresh, you will get access to the latest data while maintaining your session including selections and layout state.



Open URL...



Opens the **Open URL** dialog. In this dialog you type a valid URL to any web page. The web page will be opened in a separate window inside QlikView. This functionality may be used e.g. for the QlikView Publisher Accesspoint or for pages displaying QlikView Server documents via the Java or Zero-Footprint clients. Opened web pages can be accessed via the **Windows** menu just as with standard QlikView document windows.

Open Ftp...

Opens a file from an ftp server of your choice. See Figure 58 on page 239.

Close

Closes the active document. If changes have been made, you are asked whether you want to save it as a file.

Favorites

This menu allows you to create and maintain a list of favorite documents (local or on servers) for fast access independently of the **Recent Files** list (see below).

Add to Favorites

Opens the **Add to Favorites** dialog which lets you add the currently active document to the **Favorites** list. This option is only available when a document is open.



Organize Favorites

Opens the **Organize Favorites** dialog which lets you delete and rename entries in the Favorites list.

List of Documents

A list of favorite documents. Choosing one in the list is equivalent to opening the document.

Save

Saves the active document in a file. Data, script and layout are saved. This command can also be invoked via the keyboard shortcut CTRL+S.

**Save As...**

Saves the active document in a new document file under a new name. It is possible to save the QlikView file in a previous file format. This command can also be invoked via the keyboard shortcut F12.

Save Link...

Saves a link to a document opened on a QlikView Server as a text file on the local machine. The file will have the qvw extension but will not contain any data or layout. When opening such a link document QlikView will attempt to reconnect to the server and open the document on the QlikView Server. This command is not available for local documents.

**Mail as Attachment**

Only available when working with a local document. Creates an e-mail with a copy of the current qvw document attached. The mail recipient will be able to open the qvw document provided he has access to QlikView and access rights to the document (if section access security is used). You have to have an e-mail client configured for this command to work.

Mail with Bookmark as Link...

Only available when working with a QlikView Server document. Creates an e-mail with a URL link to the current server document. A temporary server bookmark will be created (including layout state) and encoded in the URL. The mail recipient will be able to use the URL link to open the server document and see what you see, provided that he has access rights to the document and its data. You need to have an e-mail client configured for this command to work. The QlikView Server must be configured to allow server bookmarks.

Print...

Opens the standard **Print** dialog box (see page 261 in Book II), allowing you to print the current sheet object. This command is not available for list boxes. To print the contents of a list box, use the command **Print Possible**. This command can also be invoked via the keyboard shortcut CTRL+P.



Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+P.



Print Possible...

Opens the standard **Print** dialog box (see page 261 in Book II), allowing you to print the possible (selected and optional) values of the current list box. This command is only available for list boxes.



Print Possible as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected, allowing you to print the possible (selected and optional) values of the current list box. After hitting **OK** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately. This command is only available for list boxes. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+P.



Print Sheet...

Opens the **Print** dialog (see page 269 in Book II), allowing you to print the current sheet or all the sheets of the document.

Print Preview...

Opens a dialog showing a print preview of the active object (if printable). See further on page 269 in Book II.



Export

This menu has the following options:

Export Contents...

This option is only available when an exportable sheet object is active. It then performs the export operation found on the sheet object's **Object** menu.

Export Sheet Image...

Opens a dialog for saving an image of the current sheet to file. The image can be saved as bmp, jpeg, gif or png.

Export Document Layout

Opens a dialog for saving the document layout as an XML file. No data from the document are saved in the XML file.

Export Sheet Layout

Opens a dialog for saving the current sheet's layout as an XML file. No data from the document are saved in the XML file.

Import**Import Document Layout**

Opens a dialog for importing the document layout of another document saved as XML. No data are imported.

Import Sheet Layout

Opens a dialog for importing the layout from another sheet saved as XML. No data are imported.

Edit Script...

Opens the **Edit Script** dialog box (Figure 51 on page 217). Here you can generate and execute scripts that open one or several databases in text files, and fetch data from them. This command can also be invoked via the keyboard shortcut CTRL+E.

**Reload**

Executes the current load script and reloads data to the active QlikView document. This needs to be done if the contents of the database have been changed since the last time a reload was made. This command can also be invoked via the keyboard shortcut CTRL+R.

**Partial Reload**

Executes the current load script, including all script commands, such as **drop table**, and reloads data to the active QlikView document. However, only those tables whose **load** and **select** statements are preceded by the **replace** (see page 326) or **add** (page 284) prefix are reloaded. Data tables that are not affected by this kind of **load** or **select** statements will not be affected by the partial reload. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+R.

**Reduce Data**

Opens a menu containing the two following commands:

Keep Possible Values

Reduces the QlikView database by removing all excluded values.



Remove All Values

Creates a template by removing all the values from the QlikView database, but keeping the database structure and the layout.

Table viewer...

Opens the **Table Viewer** dialog (see page 228) where the structure of the loaded data can be studied in a graphical view of tables, fields and associations. This command can also be invoked via the keyboard shortcut CTRL+T.



Recent Documents

A listing of the last documents opens. The number of documents to be shown can be set in the **User Preferences** dialog. The default number is 8. Choosing one in the list is equivalent to opening the document.

Exit

Closes the open documents and exits QlikView.

8.2 Edit Menu

The **Edit** menu contains the following commands :

Undo Layout Change

Undoes the last layout action. Changes that can be undone includes all moving, sizing and removing of sheet objects as well as changes to sheet object properties. Removing sheets, changes to sheet properties and changes to document properties can also be undone. The **Undo** command can be repeated. Certain operations, e.g. **Reload** and **Reduce Data** will empty the Undo/Redo buffer. This command can also be invoked via the keyboard shortcut CTRL+Z.



Redo Layout Change

Redoes the latest undone layout action. The Redo command can be repeated for as long as there are undone actions to redo. Certain operations, e.g. **Reload** and **Reduce Data** will empty the Undo/Redo buffer. This command can also be invoked via the keyboard shortcut CTRL+Y.



Cut

Removes the selected sheet object(s) from the sheet and puts it(them) on the clipboard for pasting elsewhere in the QlikView document. If only one sheet object is active, the



image of this object will also be put on the clipboard. This command can also be invoked via the keyboard shortcut CTRL+X.

Copy

Copies the selected sheet object(s) to the clipboard for pasting elsewhere in the QlikView document. If only one sheet object is active, the image of this object will also be put on the clipboard. This command can also be invoked via the keyboard shortcut CTRL+C.



Paste

Pastes a sheet object or several sheet objects which have previously been put on the clipboard via the **Cut** or **Copy** commands within the QlikView document. This command can also be invoked via the keyboard shortcut CTRL+V.



Format Painter

Copies formatting from one sheet objects to one or many other sheet objects.



Remove

Removes selected sheet object(s). This command can also be invoked via the keyboard shortcut DEL.

Activate All

Activates all the sheet objects on the active sheet. This command can also be invoked via the keyboard shortcut CTRL+A.

Search

Opens the text search window if a list box or opened multi box is active. Read more about how to use **Search** on page 130. This command can also be invoked via the keyboard short CTRL+F.



Fuzzy Search

Opens the text search window in fuzzy search mode provided that a list box or opened multi box is active. Read more about how to use **Fuzzy Search** on page 131.



Advanced Search

Opens the **Advanced Search** dialog, provided that a list box or opened multi box is active. The dialog makes it possible to enter advanced search expressions. This command can also be invoked via the keyboard shortcut SHIFT+CTRL+F.

Copy Mode

Switches from logic mode to copy mode. Values clicked while in copy mode are copied to the Clipboard without changing the logical state of the QlikView document.



8.3 View Menu

The **View** menu contains the following commands :

Sheets

Opens a menu with all the sheets currently visible in the document. The active sheet is checked. Select a sheet to activate it.

Toolbars

Opens a menu with the different toolbars currently available. It is possible to design your own toolbars, but the following five are supplied as standard:

Standard

Toggles the main toolbar on or off.

Navigation

Toggles the navigation toolbar on or off.

Design

Toggles the design toolbar on or off.

Collaboration Pane

Toggles the Collaboration Pane on or off. See “The Collaboration Pane” on page 255 in book II for further information on **Collaboration Pane**.

Sheets

Toggles the sheet toolbar on or off.

Bookmark

Toggles the bookmark toolbar on or off.

Customize...

Opens the **Customize** dialog, which helps you customize your toolbars.

Statusbar

Shows a statusbar at the bottom of the QlikView application windows. The text in the bar contains the date and time the application was last reloaded.

Zoom

Opens a menu containing a number of zoom factors from 25% to 400%. The zoom factor is applied on the current sheet only. Zoom

factors other than those listed can be applied in the **General** page of the **Sheet Properties** dialog.

Resize Window

This command offers a menu for resizing of the QlikView application window to one of several common screen resolutions.

Fit Zoom to Window

Adjusts the zoom factor of the current sheet to fit all sheet objects into the current window frame.

Apply Zoom to All Sheets

Applies the zoom factor of the current sheet to all sheets in the document.

Design Grid

Toggles the design grid, sheet object placeholders for active object(s) and snap-to-grid for sizing and moving objects in the layout on or off. This command can also be invoked via the keyboard shortcut CTRL+G.

Current Selections

Toggles the **Current Selections** dialog on or off. This command can also be invoked via the keyboard shortcut CTRL+Q.



8.4 Selections Menu

The **Selections** menu contains the following commands:

Back

QlikView remembers the last 100 selections. By clicking this button, you go one step back in the list of selections. This command can also be invoked via the keyboard shortcut SHIFT+LEFT ARROW.



Forward

By clicking **Forward**, you go one step forward in the list of selections (equivalent to cancelling the last **Back** command). See **Back** above. This command can also be invoked via the keyboard shortcut SHIFT+RIGHT ARROW.



Lock

Locks all the selected cells. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+L.



Unlock

Unlocks all the locked cells. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+U.



Clear

Applies the start selection of a QlikView document, which can be configured, see **Set Clear State** below. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+D.



Clear All

Clears all the current selections except the ones that are locked.

Unlock and Clear All

Clears all the current selections, including the ones that are locked.



Set Clear State

Sets the current selection as **Clear State**.

Reset Clear state

Resets **Clear States**.

8.5 Layout Menu

The **Layout** menu contains the following commands :

Add Sheet

Adds a tabbed sheet in which you can display a new set of sheet objects.

Provided that both client and server are QlikView version 9 or later it is possible to add a new sheet to the layout even if working with a document on QlikView Server.



Promote Sheet

Moves the active tab one step to the left.



Demote Sheet

Moves the active tab one step to the right.



Remove Sheet

Removes the active sheet.



Select Fields...

Opens the **Fields** page in the **Sheet Properties** dialog box. Here you can select the fields to be displayed on the current sheet. This command is not available when working with documents on QlikView Server.



Server Objects

Opens the **Server Objects** dialog box (see page 257). Here you can manage personal and share server objects. This menu option is only available when working with a document on QlikView Server and

only if the server and the server document have been configured to allow server objects.

New Sheet Object

Opens a menu in which you can choose to create one of the different sheet objects. When created you will be presented with the corresponding **Properties** dialog for configuring the new object.

Provided that both client and server are QlikView version 8 or later it may be possible to add this new sheet object to the layout even if working with a document on QlikView Server. This requires a) that you have a license, b) that the server document has been configured to allow server objects and c) that the QlikView Server is configured to allow server objects.

Rearrange Sheet Objects

Automatically arranges the sheet objects on the active sheet.



Adjust Object Maximum Size to Current Size

Some types of sheet objects, like list boxes, multi boxes and tables, may have a larger maximum size than what is currently utilized. This allows e.g. a table to expand on the sheet when more data is added. This operation resets the maximum object size to the currently used size for all selected objects on the active sheet. You may use CTRL+A to select all sheet objects on the sheet. It is recommended to always perform this operation on all list boxes, multi boxes, table boxes and table charts in the document before using the **Objects Client Page Generator wizard**.



Adjust Off-Screen Objects

Moves all sheet objects with a position outside the current QlikView window to a temporary position inside the visible area. Unless the objects are explicitly moved from their temporary positions, the original positions will be maintained when the document is saved.



Align/Distribute

Under this menu you will find a number of commands for aligning and distributing sheet objects in the layout.

Left Align

Aligns the active sheet objects along their left border.



Center Horizontally

Aligns the active sheet objects along their centers on the horizontal axis.



Right Align

Aligns the active sheet objects along their right border.



Bottom Align

Aligns the active sheet objects along their bottom border.



Center Vertically

Aligns the active sheet objects along their centers on the vertical axis.



Top Align

Aligns the active sheet objects along their top border.



Space Horizontally

Distributes the active sheet objects on the horizontal axis with equal spaces between them.



Space Vertically

Distributes the active sheet objects on the vertical axis with equal spaces between them.



Adjust Top

Arranges the active sheet objects from the horizontal top edge of the topmost object and downwards with minimal spaces between them.



Adjust Left

Arranges the active sheet objects from the vertical edge of the leftmost object and to the right with minimal spaces between them.



8.6 Settings Menu

The **Settings** menu contains the following commands:

User Preferences...

Opens the **User Preferences** dialog, which contains settings that the user normally does not change when switching to another document. This command can also be invoked via the keyboard shortcut CTRL+ALT+U.



Document Properties...

Opens the **Document Properties** dialog, which contains settings concerning the entire document. Some common proper-



ties for the sheet objects in the document can also be set in this dialog. This command can also be invoked via the keyboard shortcut CTRL+ALT+D.

Sheet Properties...

Opens the **Sheet Properties** dialog, which contains settings concerning the current sheet. Some common properties for the sheet objects on the sheet can also be set in this dialog. This command can also be invoked via the keyboard shortcut CTRL+ALT+S.



Variable Overview...

Opens the **Variable Overview** dialog (see page 199) where all non-hidden variables and their values are shown in a single list. This command can also be invoked via the keyboard shortcut CTRL+ALT+V.



Expression Overview...

Opens the **Expression Overview** dialog (see page 201) where all document, sheet and sheet object expressions can be shown and centrally maintained in a single list. This command can also be invoked via the keyboard shortcut CTRL+ALT+E.



8.7 Bookmarks Menu

The current state of selections can be saved as bookmarks for later use. Bookmarks may be document bookmarks, stored with the document or personal bookmarks, stored separately on the user's computer. The **Bookmarks** menu contains the following commands:

Document Bookmarks

A list of the first ten available document bookmarks in the active document. Click on a bookmark name to apply it.

My Bookmarks

A list of the first ten available personal bookmarks linked to the active document. Click on a bookmark name to apply it.



Add Bookmark

Opens the **New Bookmark Object** dialog for saving the current set of selections as a bookmark. The default name can be changed in the **New Bookmark Object** dialog that opens automatically. This command can also be invoked via the keyboard shortcut CTRL+B.



Replace Bookmark

Opens a menu with the ten first document bookmarks currently defined in the document, followed by the ten first personal bookmarks for the document. By choosing one of these, the contents of that bookmark will be replaced with the current state of selections and variable values.

Remove Bookmark

Opens a menu with the ten first document bookmarks currently defined in the document, followed by the ten first user bookmarks for the document. By choosing one of these, that bookmark will be deleted.

More...

Opens the **Bookmarks** dialog containing detailed information about all the saved bookmarks. From this dialog you can also delete and select bookmarks, set their internal display order and view additional information. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+B



Import...

Opens the **Import Bookmark(s)** dialog for importing bookmarks from a QlikView bookmark file.

Export...

Opens the **Export Bookmark(s)** dialog for exporting bookmarks to a QlikView bookmark file.

For more information about bookmarks, see page 149.

8.8 Reports Menu

The **Reports** menu lists all available existing reports for immediate selection. Reports may be document reports, stored with the document or personal reports, stored separately on the user's computer. The **Reports** menu contains the following commands:

Document Reports

A list of all document reports in the active document. Clicking one of the report names opens the **Print** dialog for printing of the report. The list of reports will be greyed out if there is no printer installed on the server/workstation.

My Reports

A list of all user reports in the active document. Clicking one of the report names opens the **Print** dialog for printing of the report. The

list of reports will be greyed out if there is no printer installed on the server/workstation.

Edit Reports...

Opens the **Edit Reports** dialog (see page 162) for creating new or edit existing reports. From this dialog you can also delete and select reports, design their layout, add additional pages and images etc.



8.9 Tools Menu

The **Tools** menu contains the following commands:

Edit Module...

Opens the **Edit Module** dialog where you can create macros (see page 313). This command can also be invoked via the keyboard shortcut CTRL+M.



Open QlikView AccessPoint

Opens the QlikView AccessPoint in a html window inside QlikView. This command is only available if you have configured a URL for QlikView AccessPoint on the **Locations** page of the **User Preferences** dialog.

Open QlikView Management Console

Opens the QlikView Management Console in a html window inside QlikView. This command is only available if you have configured a URL for QlikView Management Console on the **Locations** page of the **User Preferences** dialog.

Open QlikView Enterprise Management Console

Opens the QlikView Enterprise Management Console in a html window inside QlikView. This command is only available if you have configured a URL for QlikView Enterprise Management Console on the **Locations** page of the **User Preferences** dialog.

Objects Clients Page Generator

Opens a dialog that helps you generate html pages for displaying the current document on QlikView Server for one or more of the three object based clients. See page 285 in Book II.

Quick Chart Wizard...

Opens the **Quick Chart** dialog where you can create bar, line and pie charts with only very basic settings.



Time Chart Wizard...

The time chart wizard helps you to build charts where a given measure (expression) should be qualified and often compared by different time periods, e.g. current year, last year, year-to-date etc.



Statistics Chart Wizard...

The statistics chart wizard provides guidance for those who want to apply common statistical tests on data in QlikView.

Box Plot Wizard

The **Box Plot Wizard** helps you with the task of defining a box plot chart. See further description on page 261 in Book III.

Alerts...

Opens the **Alerts** dialog (see page 183) where alerts can be defined and edited. This command can also be invoked via the keyboard shortcut CTRL+ALT+A.



Alert Wizard...

The alert wizard helps you with the task of defining an alert. See further description on page 189.

Theme Maker Wizard

Invokes the **Theme Maker Wizard** for creation of a new theme or editing an existing theme. See chapter 20 in Book II for details on QlikView layout themes.

8.10 Object Menu

The **Object** menu is an object menu for the sheet object that is currently active. It can be opened by choosing **Object** on the menu bar, or by clicking with the right mouse button on the sheet object. For a description of the different object menus, see the section on the specific object.

8.11 Window Menu

The **Window** menu contains the following commands:

Cascade

Arranges windows in a cascade so that they overlap.

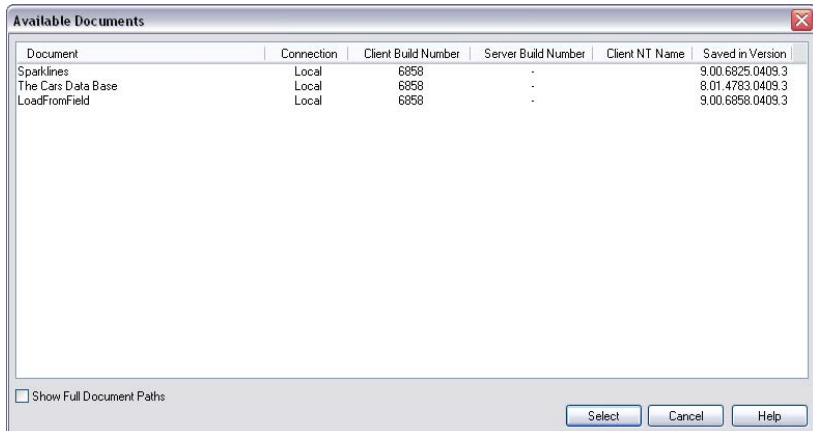
Tile

Arranges windows as non-overlapping tiles.

Arrange Icons

Arranges icons at the bottom of the window.

Available Documents...



Opens the **Available Documents** dialog. This dialog can be used for selecting the active document when you have more than ten documents open at the same time. Select a document in the list and click on **Select** to make that document the active window. Clicking on **Cancel** closes the dialog without changing the active window.

Document

Name of the qvw document. If the check box **Show Full Document Paths** at the bottom of the dialog is marked the document name will be given with its full file path.

Connection

Local (document opened on local computer) or Remote (document opened on QlikView Server).

Client Build Number

Build number of QlikView client.

Server Build Number

Build number of QlikView Server for remote documents.

Client NT Name

Windows NT authenticated identity of client user when connection is made with NT authentication.

Saved in Version

Full information about the QlikView version that last saved the document (available for local documents only).

Also, an alphabetic listing of the ten first open document windows is displayed for quick selection. Selecting a document from the list makes it the active window.

8.12 Help Menu

The **Help** menu contains the following commands:

Contents

Opens the QlikView help file.



Using Help

Displays instructions on how to use the help file.

Show Start Page

Shows the start page each time you start QlikView. If you don't want the start page to appear when you start the program, deselect the **Show Start Page When Launching QlikView** check box in the left lower corner of the start page.

QlikTech on the Web

Opens a menu containing the following commands:

Support

If your computer is connected to the Internet, this command will take you directly to the Support section of the QlikTech home page. From here you can access the FAQ (Frequently Asked Questions) section. If you are in need of further support, this page also lets you contact QlikTech via e-mail.

What's New

If your computer is connected to the Internet, this command will take you directly to the What's New section of the QlikTech home page.

QlikTech Home Page

If your computer is connected to the Internet, this command will take you directly to the QlikTech home page.

QlikView Update...

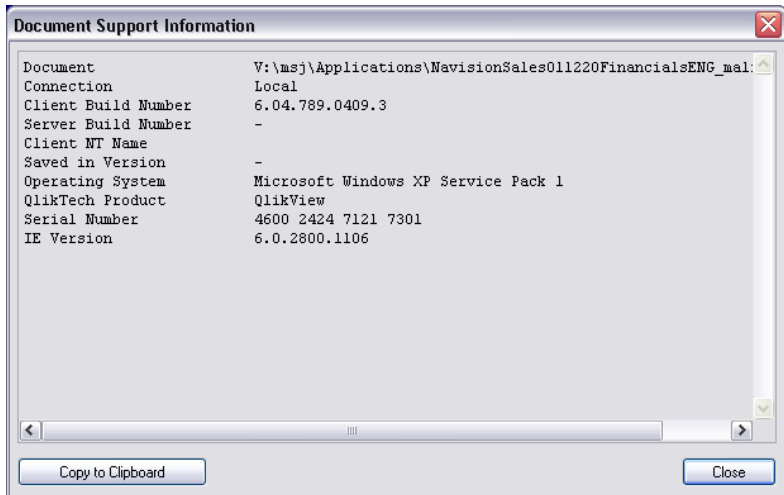
Opens the **QlikView Update** dialog from which you can make QlikView contact the QlikView update server to see if any program updates are available. You can also specify automatic checks for updates every *n* days. In order to use this feature you must be connected to the Internet. As part of the procedure your QlikView registration information as well as information about installed QlikView version and operating system version will be sent to the QlikView

update server. This feature may be permanently disabled by the system administrator in certain environments.

License Update...

Opens the **License Enabler File Editor** dialog from which you can make QlikView contact the QlikView license server to download license registration info. This is normally done automatically, but in some cases it may be necessary to do it manually. If you have no connection to the Internet you will receive your License Enabling File via mail. In this case this is where you can paste the data into QlikView.

Document Support Info...



Opens the **Document Support Information** dialog. This dialog shows a list of support data for the active document. The information can and should be used whenever contacting QlikTech support and when reporting bugs or problems in relation to QlikView documents. Note that some rows are only relevant for local documents while some refer only to remote documents. This dialog can also be invoked via the keyboard shortcut CTRL+SHIFT+Q.

About QlikView...

Opens the **About** dialog box which displays the QlikView version, the serial number and the name of the owner.



9 THE TOOLBARS AND THE STATUS BAR

9.1 The Toolbars

QlikView has five standard toolbars and one menubar. The standard toolbar normally contains buttons for tasks that you perform while using QlikView documents, whereas the design toolbar normally contains buttons for tasks you perform when creating or changing the layout of a document. The navigation toolbar contains the most frequently used commands for logical operations in a document. The sheet toolbar offers an alternative method for navigating different sheets whereas the bookmark toolbar offers an alternative way of accessing bookmarks.

Each of the toolbars can be individually turned on and off. All toolbars are completely customizable and may contain any of the available command buttons.

All toolbars can be moved around by pointing at them on the dotted line to their far left. Hold down the left mouse button and drag to any position you like. The toolbars can be docked to any side of the QlikView application window.

9.2 The Standard Toolbar

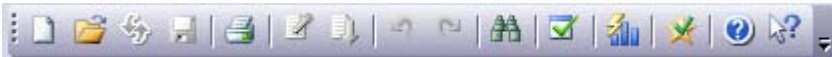


Figure 6. The standard toolbar

The QlikView standard toolbar (see above) contains buttons for the most frequently needed functions. To toggle the standard toolbar on or off choose **Standard Toolbar** under **Toolbars** on the **View** menu. The figure above and the text below refer to the default contents of the standard toolbar.

New

Creates a new document. When pressed, it opens an empty QlikView screen. This command can also be invoked via the keyboard shortcut CTRL+N.



Open

Opens an existing document file, either a QlikView file or a text file. If you open a text file, you will automatically get to the Table file wizard, which analyzes its contents and helps you create a script.

This command can also be invoked via the keyboard shortcut CTRL+O.

Refresh

This command is only available with documents opened on QlikView Server and when there is a newer version of the document available on the server. When invoking a refresh, you will get access to the latest data while maintaining your session including selections and layout state.



Save

Saves the active document as a file. The default file format is set in the **User Preferences** dialog, **Save** page (Figure 10.2 on page 97). This command can also be invoked via the keyboard shortcut CTRL+S.



Print

Prints the active object. This command can also be invoked via the keyboard shortcut CTRL+P



Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This button is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.



Edit Script

Opens the **Edit Script** dialog (Figure 51 on page 217). This command can also be invoked via the keyboard shortcut CTRL+E.



Reload

Reloads data to the QlikView document by executing the load script. This command can also be invoked via the keyboard shortcut CTRL+R.



Undo Layout Change

Undoes the last layout action. Changes that can be undone include all moving, sizing and removing of sheet objects, as well as changes to sheet object properties. Removing sheets, changes to sheet properties and changes to document properties can also be undone. The **Undo** command can be repeated. Certain operations, e.g. **Reload** and **Reduce Data**, will empty the Undo/Redo buffer. This command can also be invoked via the keyboard shortcut CTRL+Z.



Redo Layout

Redoes the last layout action. The **Redo** command can be repeated for as long as there are undone actions to redo. Certain operations, e.g. **Reload** and **Reduce Data**, will empty the Undo/Redo buffer. This command can also be invoked via the keyboard shortcut CTRL+Y.



Search

Opens the text search window if a list box, opened drop-down select in a table box or opened multi box is active. The action is equal to that of just start typing in an opened list box or multi box. This command can also be invoked via the keyboard shortcut CTRL+F.



Current Selections

Opens the **Current Selections** dialog in which it is possible to see the selections that are active. This command can also be invoked via the keyboard shortcut CTRL+Q.



Quick Chart

Opens the **Quick Chart** dialog, where you can create a bar chart, line chart or pie chart using only the most basic settings.



Add Bookmark

Saves the current set of selections as a bookmark. In the **Create Bookmark** dialog that opens, you can specify a name for the bookmark. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+B.



Help Topics

Opens the QlikView help program.



Context Help

Displays specific help concerning the chosen object, e.g. a menu command. After clicking this button, simply move the question mark to the object on which you need help.



9.3 The Navigation Toolbar



Figure 7. The Navigation toolbar

The QlikView navigation toolbar (see above) contains buttons for the most frequently needed functions when using QlikView for analyzing data. To toggle the navigation toolbar on or off choose **Navigation Toolbar** under **Toolbars** on the **View** menu. The figure above and the text below refer to the default contents of the **Navigation Toolbar**.

Clear

Clicking on this button applies the start selection of a QlikView document, which can be configured, see **Set Clear State** below. The drop-down menu offers the following options:

Clear

The start selection of a QlikView document. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+D.

Clear All

Clears all selections, excluding locked selections.

Unlock and Clear All

Unlocks and clears all selections.

Set Clear State

Sets the current selection as **Clear State**.

Reset Clear State

Resets **Clear State** to no selections.

Back

QlikView remembers the last 100 selections. By clicking this button, you go one step back in the list of selections. This command can also be invoked via the keyboard shortcut SHIFT+LEFT ARROW.

Forward

By clicking **Forward**, you go one step forward in the list of selections (equivalent to cancelling the last **Back** command). See **Back** above. This command can also be invoked via the keyboard shortcut SHIFT+RIGHT ARROW.

Lock

Prevents selections from being cleared by mistake. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+L.

Unlock

Takes away the lock described above. This command can also be invoked via the keyboard shortcut CTRL+SHIFT+U.

9.4 The Design Toolbar



Figure 8. The design toolbar

The QlikView design toolbar (see above) contains buttons for tasks you perform when creating or changing the layout of a document. By default, this toolbar is not displayed. To toggle the design toolbar on or off choose **Design Toolbar** under **Toolbars** on the **View** menu. The figure above and the text below refer to the default contents of the design toolbar.

Add Sheet

Adds a new sheet to the document.



Promote Sheet

Moves the active sheet one step further to the left.



Demote Sheet

Moves the active sheet one step further to the right.



Sheet Properties

Opens the **Sheet Properties** dialog, from which you can modify the active sheet.



Create List Box

Creates a list box to display the field of your choice from the database table.



Create Statistics Box

Creates a statistics box, which calculates statistical entities based on the possible values of a field.



Create Table Box

Creates a table box, suitable for showing record-oriented information.



Create Multi Box

Creates a multi box, suitable for showing different attributes.



Create Chart

Creates a chart that can be made to display fields and calculated dimensions.



Create Input Box

Creates an input box, suitable for displaying and entering data into QlikView variables.



Create Current Selections Box

Creates a current selections box, suitable for displaying the current selections directly in the layout.



Create Button

Creates a button object that performs actions in QlikView, e.g. shortcut, export etc.



Create Text Object

Creates an object for displaying text information or images.



Create Line/Arrow Object

Creates a line/arrow object, suitable for drawing a line or an arrow in the layout.



Create Slider/Calendar Object

Creates a new slider/calendar object.



Create Bookmark Object

Creates a new bookmark object.



Create Search Object

Creates a new search object.



Create Custom Object

Creates a new custom object.



Create Time Chart

The **time chart wizard** helps you to build charts where a given measure (expression) should be qualified and often compared by different time periods, e.g. current year, last year, year-to-date etc.



Format Painter

This button makes it possible to copy formatting from one sheet objects to one or many other sheet objects. In order to copy formatting to a single object, first click on the source object, then single-click on the format painter button and then click on the target object. In order to copy formatting to multiple objects, first click on the source object, then double-click on the format painter button and then click on each of the target objects. You stop the copying by clicking the button again or pressing ESC. When you copy formatting between sheet objects of different types or when you click on the caption of the target object(s), only border/caption properties will be copied. When you copy between sheet objects of



the same type, additional object type specific properties will be copied.

Align Left

Aligns the active sheet objects along their left border.



Center Horizontally

Aligns the active sheet objects along their center on the horizontal axis.



Align Right

Aligns the active sheet objects along their right border.



Align Bottom

Aligns the active sheet objects along their bottom border.



Center Vertically

Aligns the active sheet objects along their center on the vertical axis.



Align Top

Aligns the active sheet objects along their top border.



Space Horizontally

Distributes the active sheet objects on the horizontal axis with equal spaces between them.



Space Vertically

Distributes the active sheet objects on the vertical axis with equal spaces between them.



Adjust Left

Arranges the active sheet objects from the vertical edge of the left-most object and to the right with minimal spaces between them.



Adjust Top

Arranges the active sheet objects from the horizontal top edge of the topmost object and downwards with minimal spaces between them.



Document Properties

Opens the **Document Properties** dialog, from which you can modify the settings of the current document.



User Preferences

Opens the **User Preferences** dialog, where you can modify settings concerning the way the user works.



Edit Module

Opens the **Edit Module** dialog where macros and custom defined functions can be written in VBScript or JScript.



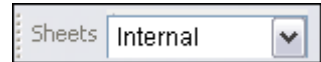
Table Viewer

Opens the Table Viewer dialog where the data table structure is displayed.

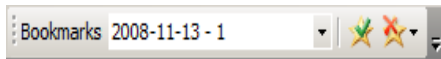


9.5 The Sheet Toolbar

The sheet toolbar contains a drop-down list of all sheets in the document and offers an alternative way of changing sheets. The name of the active sheet is always shown in the drop-down box. To toggle the **Sheet** toolbar on or off choose **Sheet Toolbar** under **Toolbars** on the **View** menu.



9.6 The Bookmark Toolbar



The bookmark toolbar contains a drop-down list of all bookmarks in the document and offers an alternative way of changing bookmarks. After selecting a bookmark, the name of the bookmark will be shown in the drop-down box until selections or variable values are changed.

Add Bookmark

Saves the current selection as a bookmark.



Remove Bookmark

Removes the bookmark selected in the drop-down menu.



9.7 Configuration of the Toolbars

All toolbars are customizable, which means that you can configure them to display the buttons you use the most. Select **Customize...** under **Toolbars** in the **View** menu to open the **Customize** dialog. The dialog has three pages.

Note Customization of toolbars can be locked by means of registry entries.

Toolbars

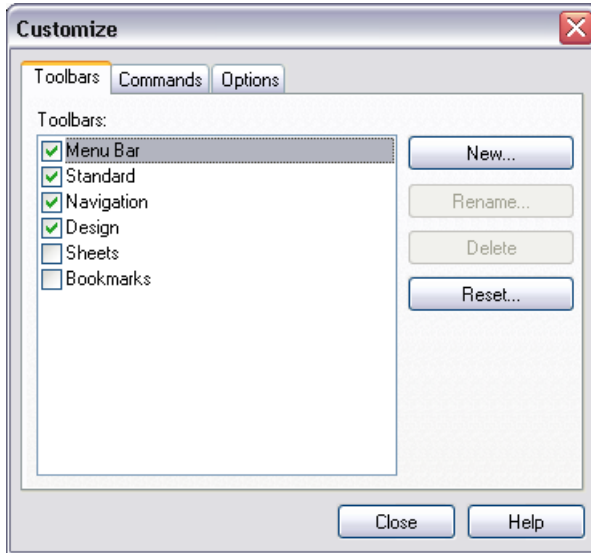


Figure 9. The Toolbars page of the Customize dialog

This page contains a list of all available toolbars and the menu bar. You can turn toolbars on or off by selecting/deselecting them in the list.

New...

Click this button to create your own new toolbar.

Rename...

Click this button to rename the selected toolbar. The command is not available for the five default toolbars.

Delete

Click this button to delete the selected toolbar. The command is not available for the five default toolbars.

Reset...

Click this button to Reset the configuration of the selected toolbar to default.

Commands

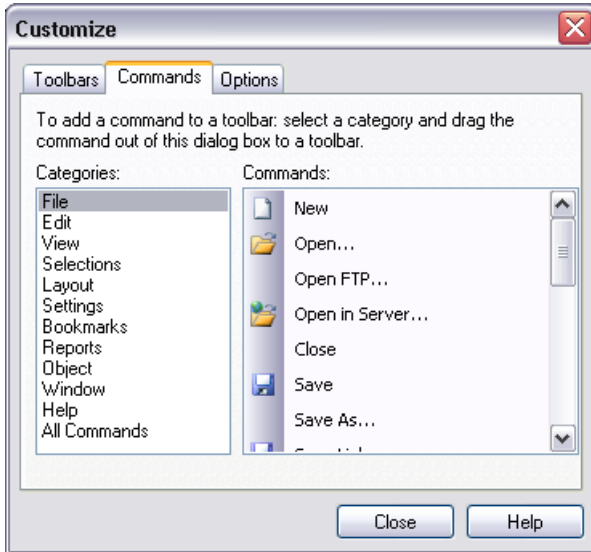


Figure 10. The Commands page of the Customize dialog

This page contains a list of all available commands. You may filter the list by choosing a category in the left column. Point at a command and drag it onto any toolbar in any position you choose.

Options

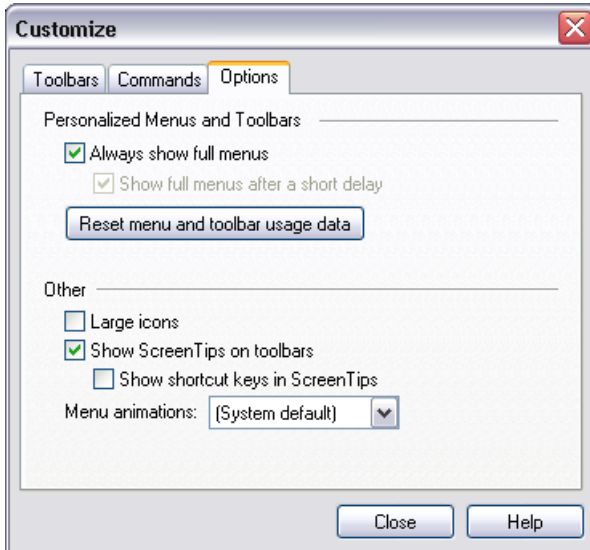


Figure 11. The Options page of the Customize dialog

This page contains additional options for toolbars.

The **Personalized Menus and Toolbars** section allows you to use shorter menus with only the most common commands.

Always show full menus

Deselect this check box to use shorter menus with only the most common commands.

Show full menus after short delay

When short menus are used, mark this check boxcheck box to have the full menus appear after hovering over the menu for some time.

Reset menu and toolbar usage data

This command deletes the records of the commands used in an application and restores the default set of visible commands to the menus and toolbars. It does not undo any explicit customizations.

The **Other** section contains the following settings:

Large icons

Mark this check box to get large icons in all toolbars.

Show ScreenTips on toolbars

Mark this check box to display tool tips when hovering over toolbar buttons.

Show shortcut keys in ScreenTips

Mark this check box to include keyboard shortcuts (where available) in the tool tips.

Menu animations

This drop-down offers a choice of menu animation schemes.

Customizing QlikView Toolbars in Large Deployments

Enabling and disabling interactive toolbar customization

QlikView 7 introduced fully customizable toolbars and menus. The interactive customization can be turned on and off by means of the two registry keys:

*HKEY_CURRENT_USER\Software\QlikTech\QlikView\Settings
7\AllowCustomizeToolbars*

and

*HKEY_CURRENT_USER\Software\QlikTech\QlikView\Settings
7\AllowCustomizeMenubar*

For the QlikView for Internet Explorer plug-in client the corresponding registry key is

*HKEY_CURRENT_USER\Software\QlikTech\QlikViewPlugin\Set-
tings 7\AllowCustomizeToolbars*

Setting the value to 1 enables interactive customization whereas the value 0 prevents it.

Enforcing toolbar settings on large numbers of computers

If you want to duplicate the toolbar settings between computers, do as follows:

- 1 Customize the toolbars on one computer
- 2 Export the following registry keys from Regedit

Ordinary QlikView:

HKEY_CURRENT_USER\Software\QlikTech\QlikView\Toolbar 7

Plug-in:

HKEY_CURRENT_USER\Software\QlikTech\QlikViewPlugin

- 3 Run the resulting reg files on target computers

If already running, QlikView has to be restarted on the target computer for the changes to take effect.

Note The QlikView versions of the source and target computers should be the same in order to guarantee 100% functionality.

9.8 The Status Bar

Below the sheet, the status bar is found. Choose **View** and **Statusbar** to toggle it on or off. Some interesting information is displayed here:

On the left side of the status bar, several things can be shown: the text **Ready** may be shown when QlikView is ready for selections; if the cursor is moved over a graphical chart, the coordinates are shown.

It is also possible to obtain help to the left on the status bar. When clicking a command or a button without releasing the mouse button, help is displayed. If the mouse cursor is moved outside the command or the button before the mouse button is released, the command will not be executed.

In the middle of the status bar a time stamp is displayed. It shows when the last reload of data was performed.

An **AND**-indicator is shown if the active object is in **and** mode.

On the right side of the status bar, the number of distinct optional (or selected) values over the total number of distinct values in the active list box is presented, preceded by a D.

Further to the right, preceded by an F, the frequency of the active field is presented, showing the number of records in the table where the field first occurs over the total number of records.

Finally a selection indicator is shown on the status bar. It will be green if selections are made that cannot be seen on the current sheet.



10 USER PREFERENCES

User preferences are settings that concern the way the user works and that are stored on the computer, not in the document file.

The **User Preferences** dialog is opened from the **Settings** menu. When the preferences are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

10.1 General

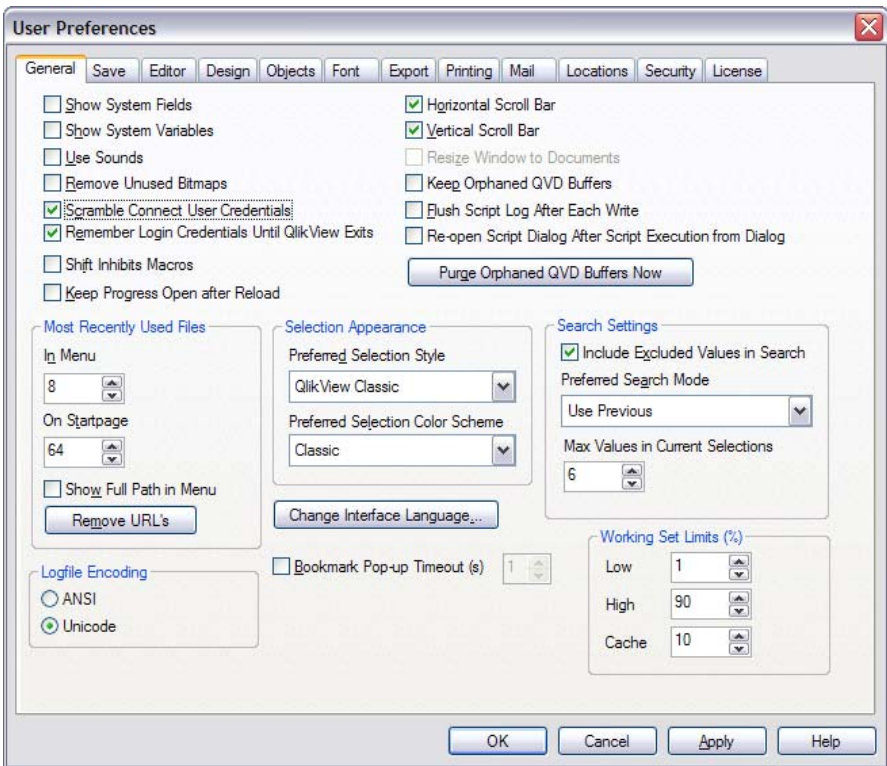


Figure 12. The General page in the User Preferences dialog

Show System Fields

Check this box to show the system fields in dialog field lists.

Show System Variables

Check this box to show the system variables in dialog variable lists.

Use Sounds

Adds sound effects to QlikView events.

Remove Unused Bitmaps

Bitmaps used in sheets and sheet objects are normally retained in the document even if their use is turned off. If this check box is marked, they will however be removed from the document when settings in a **Properties** dialog render them unused.

Scramble Connect User Credentials

This check box should be marked if you want the **connect** statement wizard to create connect statements with scrambled user ID and password (see page 290).

Remember Login Credentials Until QlikView Exits

QlikView will normally remember the user ID and password for a document after a successful login for the remainder of the QlikView session. This means that if you close the document and reopen it without closing QlikView, no new login will be required. By deselecting this check box you will force QlikView to ask for login every time the document is opened.

Shift Inhibits Macros

If this check box is selected, you may inhibit the execution of any OnOpen macro by keeping the SHIFT key depressed while opening a QlikView document.

Keep Progress Open after Reload

Marking this check box will keep the **Script Execution Progress** dialog open after the script has finished reloading.

Horizontal Scroll Bar

If this check box is selected, a horizontal scroll bar will appear in the sheet whenever any of the sheet objects cannot be fully accommodate horizontally inside the application window.

Vertical Scroll Bar

If this check box is selected, a vertical scroll bar will appear in the sheet whenever any of the sheet objects cannot be fully accommodated vertically inside the application window.

Resize Window to Documents

With this alternative checked, the size of the QlikView window is determined by the size of the document, i.e. the size of the QlikView window of the time the document was saved.

Keep Orphaned QVD Buffers

Unless this check box is marked, all automatically created QVD buffers (created via buffered prefix on **load** and **select** statements) will be removed by QlikView when the document that created them no longer exists. Each QVD buffer contains information about the complete path to the document that created it. If that path does not lead to a valid QlikView document, the QVD file will be considered orphaned. Note that this will be the case also if the QVW file has been moved or renamed. After a successful script execution (non partial reload) QlikView will scan all QVD buffers currently residing in the default folder for QVD buffers (see page 120). Any buffer found to be orphaned according to the criteria described above will be removed. By marking this check box this purge will not occur. Note that this may cause unnecessary use of hard disk space. If you decide to do it anyway, you may at all times use the **Purge Orphaned QVD Buffers Now** option to perform a manual purge.

Flush script log after each write

For performance reasons the script log is normally not written to disk after each individual statement. By selecting this check box it will. This may be useful if you use other programs to monitor the script execution via the log. However, using this setting may significantly increase script execution time, in cases where the script contains large numbers of statements.

Re-open Script Dialog After Script Execution from Dialog

If the script is executed from inside the **Edit Script** dialog and this check box is marked, the dialog will be re-opened after script execution.

Purge Orphaned QVD Buffers Now

Press this button to perform a manual purge of orphaned QVD buffers according to the rules described under **Keep Orphaned QVD Buffers** above.

Most Recently Used Files

This group is used for controlling the list of recently opened QlikView files in the **File** menu and on the **Start Page**.

In Menu

The number of files listed in the **File** menu. Default is 8.

On Start Page

The Number of files listed on the **Start Page**. Default is 64.

Show Full Path in Menu

If this box is marked the **File** menu will be widened as necessary to show full paths in the list of recently used files.

Remove URLs

By clicking this button, you remove all URLs from the list of most recently used files.

Log file Encoding

Select **ANSI** or **Unicode** for QlikView script log files.

Selection Appearance

QlikView supports a number of different ways of presenting data and making selections in list boxes and multi boxes. The **QlikView Classic**, **Corner Tag**, **LED** and **Tube** styles all use color coding for indication of selected, possible and excluded values. The **Windows Check boxes** style mimics the standard Windows interface with check boxes at each value. More information about the selection styles can be found in the chapter “Selection Styles” on page 142.

When using the selection styles based on color, there are a number of different color schemes available. The basic colors (green for selected, blue for locked etc.) cannot be changed, but variations of tone and intensity are possible.

Preferred Selection Style

Sets the default selection style. Select between the available alternatives in the drop-down list. This default can be overridden for a specific document by means of a corresponding setting on the **General** page of the **Document Properties** dialog (see page 16 in Book II).

Preferred Selection Color Scheme

Sets the default selection color scheme. Select between the available alternatives in the drop-down list. This default can be overridden for a specific document by means of a corresponding setting on the **General** page of the **Document Properties** dialog (see page 16 in Book II).

Change Interface Language

Press the button to open the **Select Interface Language** dialog. In this dialog all available languages on your computer will be listed. Select the language you prefer and restart QlikView to complete the change. The change affects both the language of the program user interface and the language of the on-line help, provided that the appropriate help files are available on your computer.

Bookmark Pop-up Time-out (s)

If this check box is marked bookmark pop-up windows will automatically close after a set number of seconds. The time delay is set in the scroll control to the right.

Search Settings

In this group settings are made for default search mode.

Include Excluded Values in Search

There are two ways to interpret text search: you either search among optional values, or you search among all values, i.e. you include the excluded values in the search.

Mark this check box to do the latter. This default value can be overridden on sheet object level.

Preferred Search Mode

The default search mode for text search in list boxes, multi boxes etc. can be set in this drop-down box. The default only applies when you start typing directly and do not use any of the menu choices or keyboard shortcuts for starting the search. The normal search mode (with ** in the search box) will be used when **Use Normal Search** is selected.

The fuzzy search mode (with ~ in the search box) will be used when **Use Fuzzy Search** is selected. When **Use Previous** is selected the search mode of the last completed search will be used. This default value can be overridden on sheet object level.

Max Values in Current Selections

Here you can specify the maximum number of distinct selected values to be shown in the current selections dialog and in the selection stamps in printouts. When more values are selected they will only be specified as 'x values of y' for the respective field.

Working Set Limits %

This control sets the minimum and maximum of the physical amount of RAM that can be used by an application.

This way it is possible to control if an application can be swapped out of physical memory or not. However, there are no guarantees that the operating system can serve the process with the amount of memory set here.

Using too high settings will degrade the performance of other processes on the computer, this may however be desirable if the computer is dedicated for QlikView.

Do not change these settings unless you are well acquainted with Windows Virtual Memory Manager! Read more about working sets in the Microsoft Windows documentation.

The settings are:

Low

sets the minimum amount of memory, in percentage, to be allocated to the application/process.

High

sets the maximum amount of memory, in percentage, to be allocated to the application/process.

Cache

sets the amount of memory, in percentage, to be used as cache for the application/process.

10.2 Save

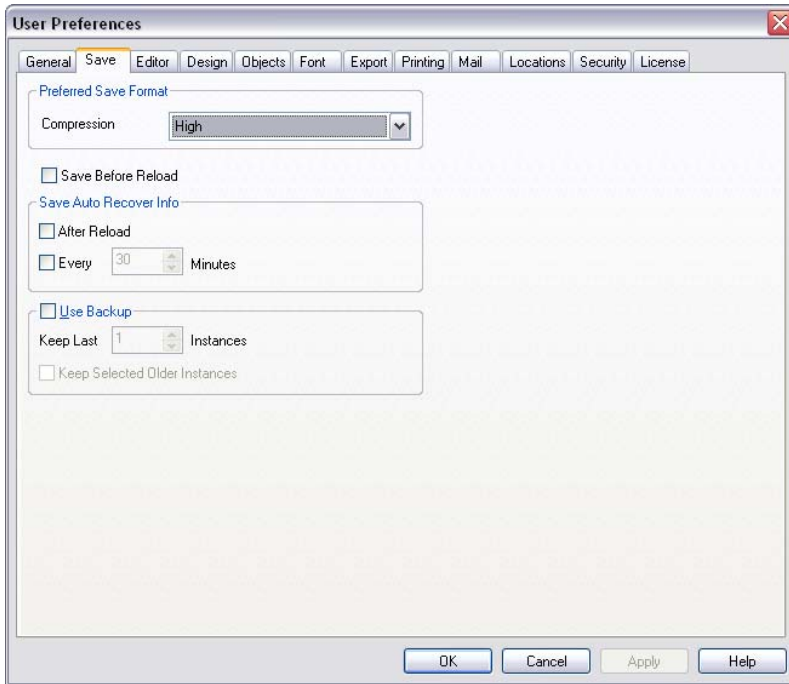


Figure 13. The Save page of the User Preferences dialog

The **Save** page contains settings for how to save QlikView documents.

Preferred Save Format

Here you can make settings with regard to the default document save format.

Compression

This drop-down list specifies the save compression mode for new documents. By using compression the size of the file will be reduced by typically 60-80% (actual results will vary with the document). When using compression, document save times will be slightly increased. With **Medium** compression all parts of the document except table data (which is already stored quite compressed inside QlikView) will undergo compression. With **High** compression (default) also table data will undergo compression, saving some space but further increasing save and load times. By

choosing **None** all data is saved without compression. See also **Save Format** settings, on **General** page of **Document Properties** (page 16 in Book II).

Save Before Reload

By checking this box the document will be automatically saved before the script is executed.

Save AutoRecover Info

In the **Save AutoRecover Info** group you can specify rules for when AutoRecover files should be saved. Such files can be used to restore lost work e.g. if the system crashes before changes have been saved. AutoRecover files are automatically deleted every time a document has been successfully saved or abandoned by the user. Every time QlikView is started a check will be made if AutoRecover files exist. If this is the case a dialog will be displayed allowing you to open (and resave) or delete these files.

After Reload

If this box is checked **AutoRecover** information will be saved after new data has been loaded with the script.

Every n Minutes

If this box is checked **AutoRecover** information will be saved every n minutes, where n can be specified in the corresponding edit box.

Note No **AutoRecover** files will be saved for a new document until it has been saved by the user under a document name.

Use Backup

If this box is checked backup copies of older versions will be retained according to the choices in the two settings below. You can specify rules for keeping copies of previous versions of the document as backup. The files are stored in the same folder as the current document and will be named "Version n of *filename*" where n is the version number (starting with 1) and *filename* is the original document name.

Keep Last n Instances

In the edit box you may specify how many backup versions will be retained of the document.

Keep Selected Older Instances

By checking this box a smart selection of older versions will be retained in addition to those kept as a result of the

setting above. The density of saved versions will decrease with age.

10.3 Editor

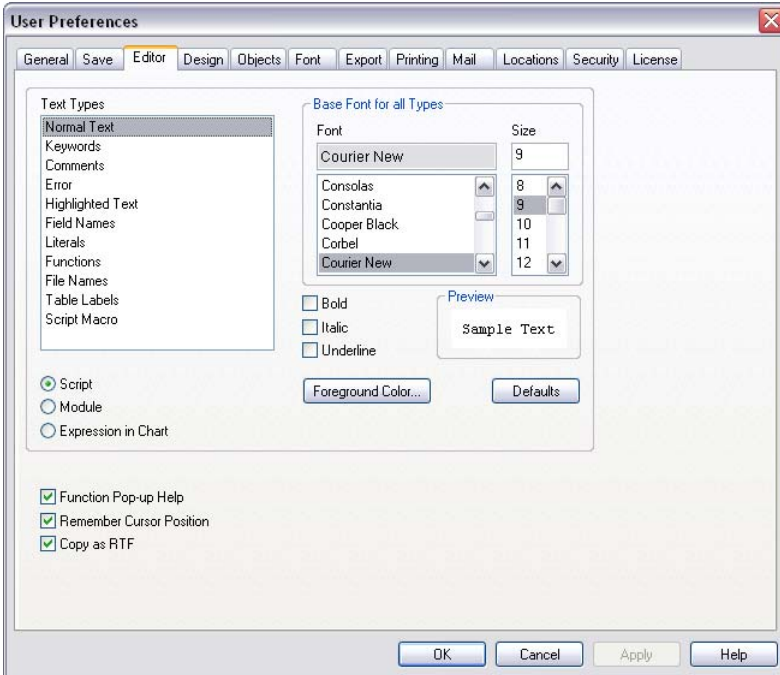


Figure 14. The Editor page in the User Preferences dialog

The **Editor** page makes it possible to customize the **Edit Script** dialog (page 217), the **Edit Module** dialog (page 313), and the **Edit Expression** dialog (page 249 in Book III). You can individually format each text type appearing in these dialogs.

Text Types

Select a text type in the list to apply a different formatting. Choose one of the options below the list to display the text types available in the **Edit Script** dialog (**Script**), the **Edit Module** dialog (**Module**) and the **Edit Expression** dialog (**Expression in Chart**), respectively.

Script

Select this option to make the text types appearing in the **Edit Script** dialog available in the **Text Types** box. The formatting of the following text types can be configured:

Normal Text

Text that does not belong to any of the categories described below.

Keywords

The keywords used in the script (described under “Script Keywords and Statements in Alphabetical Order” on page 281 and forward), e.g. **load**, **select**, **directory**, **semantic**, etc.

Comments

Comments typed in the script. For more information, see page 325.

Error

The errors that QlikView detects in the script.

Highlighted Text

When you approach a parenthesis with the cursor in the script, the parenthesis, as well as the commas enclosed by it, will be highlighted. Missing parentheses or commas will thus be easily detectable.

Field names

The names of the fields to be loaded.

Literals

Text to be loaded literally, i.e. as a text string (usually enclosed by single quotation marks). For the difference between field names and literals see page 353.

Functions

The functions used in the script (described under “Aggregation Functions” on page 359 and forward), e.g. **div**, **left**, **if**, **num**, etc.

File Names

The name of the file from which the fields are to be retrieved.

Script Macro

The variables used in the script. For more information on variables, see page 343.

Operators

The operators used in the script (described under “Operators” on page 356 and forward), e. g. **+**, **like**, etc.

Table Labels

The labels assigned to specific tables. For more information see “Table Names” on page 342.

Module

Select this option to make the text types appearing in the **Edit Module** dialog available in the **Text Types** box. The formatting of the following text types can be configured:

Normal Text

Text that does not belong to any of the categories described below.

Keywords

Visual Basic Script reserved keywords.

Highlighted Text

When you approach a parenthesis with the cursor in the script, the parenthesis, as well as the commas enclosed by it, will be highlighted. Missing parentheses or commas will thus be easily detectable.

Expression in Chart

Select this option to make the text types appearing in the **Edit Expression** dialog available in the **Text Types** box. The formatting of the following text types can be configured:

Normal Text

Text that does not belong to any of the categories described below.

Comments

Comments typed in the chart expression.

Error

The errors that QlikView detects in the expression, e.g. a missing parenthesis.

Highlighted Text

When you approach a parenthesis with the cursor in the expression, the parenthesis, as well as the commas enclosed by it, will be highlighted. Missing parentheses or commas will thus be easily detectable.

Field names

The names of the fields used.



Functions

QlikView standard function names for chart expressions.

File Names

The name of the file from which the fields are to be retrieved.

Aggregation Functions

The aggregations functions used in the expression (described under “Aggregation Functions” on page 298 and forward), e.g. **sum**, **min**, **max**, etc.

Operators

The operators used in the expression (described under “Operators” on page 295 and forward), e. g. **+**, **like**, etc.

Set Analysis

The set analysis identifiers, modifiers and operators used in the expression (described under “Set Analysis” on page 353 and forward).

Base Font for All Types

The font and the font size selected in this group will be applied to all the text types. Properties such as color, bold, italic and underline can be set individually (see below).

Bold

Makes the text bold. This setting applies to the text type selected in the list.

Italic

Displays the text in italics. This setting applies to the text type selected in the list.

Underline

Underlines the text. This setting applies to the text type selected in the list.

Preview

Gives a preview of the current settings.

Defaults

Applies the default settings.

Foreground Color

Opens a color map from which you can choose the color you wish to apply to the selected text type.

Function Pop-Up Help

If this check box is marked a pop-up help window will appear when you define functions in expressions in the script and in the **Edit**

Expression dialog. This pop-up help window describes the syntax to be used in the expression.

Remember Cursor Position

If this check box is marked the editors for script and macro will remember the position of the cursor when you leave the editor. Next time you return to the editor the cursor will be where you left it. If this feature is not used the cursor will be positioned at the start of the macro module and at the end of the script.

Copy as RTF

If this check box is marked, copy operations from the editor windows will be put on the clipboard not only as pure text but also as RTF, enabling applications supporting RTF import to paste the text with full formatting.

10.4 Design

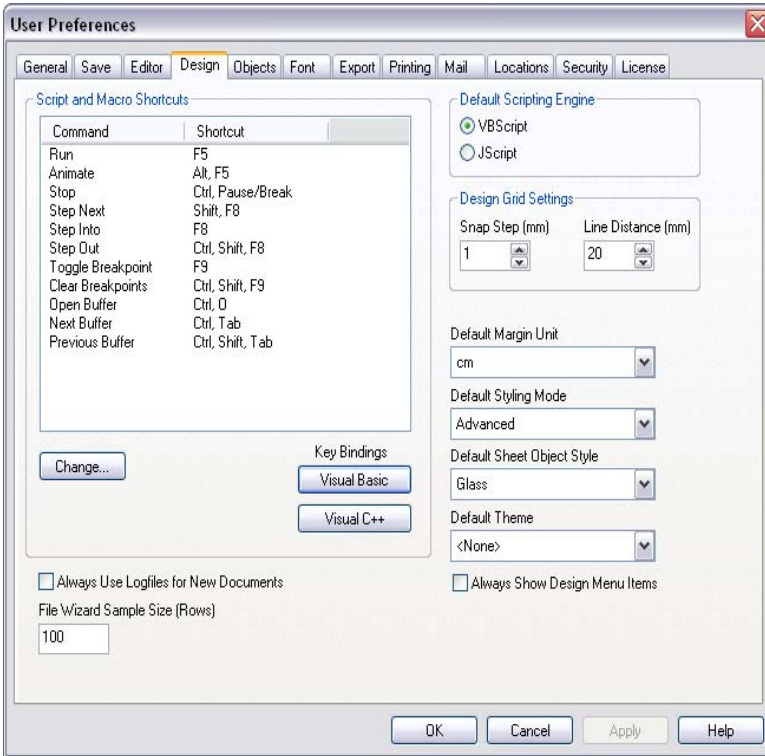


Figure 15. The Design page in the User Preferences dialog

In the **Design** page you can customize certain features related to document design.

Script and Macro Shortcuts

In the **Script and Macro Shortcuts** group you can customize keyboard shortcuts for certain actions in the script debug dialog.

Generate a list of all available keyboard shortcuts in the script by typing `ctrl+qs` in the script.

Edit

Select a command from the list and press the **Change** button to customize the keyboard shortcut for the command. Double-clicking in the list has the same effect.

Key Bindings

Visual Basic

Sets the keyboard shortcuts to defaults familiar to the Visual Basic debugging environment.

Visual C++

Sets the keyboard shortcuts to defaults familiar to the Visual C++ debugging environment.

Default Scripting Engine

Sets the default scripting engine in new documents. You may choose between **VBScript** and **JScript**.

Default Grid Settings

Sets measures for the layout design grid.

Snap Step (mm)

Sets the distance between the snap points when design grid is displayed.

Line Distance (mm)

Sets the distance between the grid lines when design grid is displayed.

Default Margin Unit

Here you may select if the default unit for margins in the **Layout** page of the **Print** dialog should be cm or inch.

Default Styling Mode

Choose one of the available modes for the object style for all your sheet objects. The mode chosen will be used as default for all new documents.

Default Sheet Object Style

Choose one of the available styles for sheet object style in this drop-down. The style selected will be used for all sheet objects in all new documents.

Always Use Logfiles for New Documents

If this box is checked, a log file will always be generated for new QlikView documents.

File Wizard Sample Size

Specifies the number of records read into the table file wizard. Normally a sample of 100 lines will be sufficient but in certain cases a higher number may be desired. Very high numbers in this setting will slow down the wizard unnecessarily and should thus be avoided.

Default Theme

Here you may select a QlikView theme which will be applied to all newly created sheets and sheet objects in all new documents. The selected theme must be accessible from disc at all times in order to be used. It is also important that the theme used is defined for all types of objects that may occur in a QlikView document. At the bottom of the drop-down list there is a **Browse...** command in case your theme resides in another location than in the default QlikView theme catalog. If no default theme is used, each new sheet object will inherit the properties of the last created or changed object.

Always Show Design Menu Items

If this check box is marked all design menu options will be available on the context menu at all times.

If unchecked some menu options will only be available when the design grid is turned on.

10.5 Objects

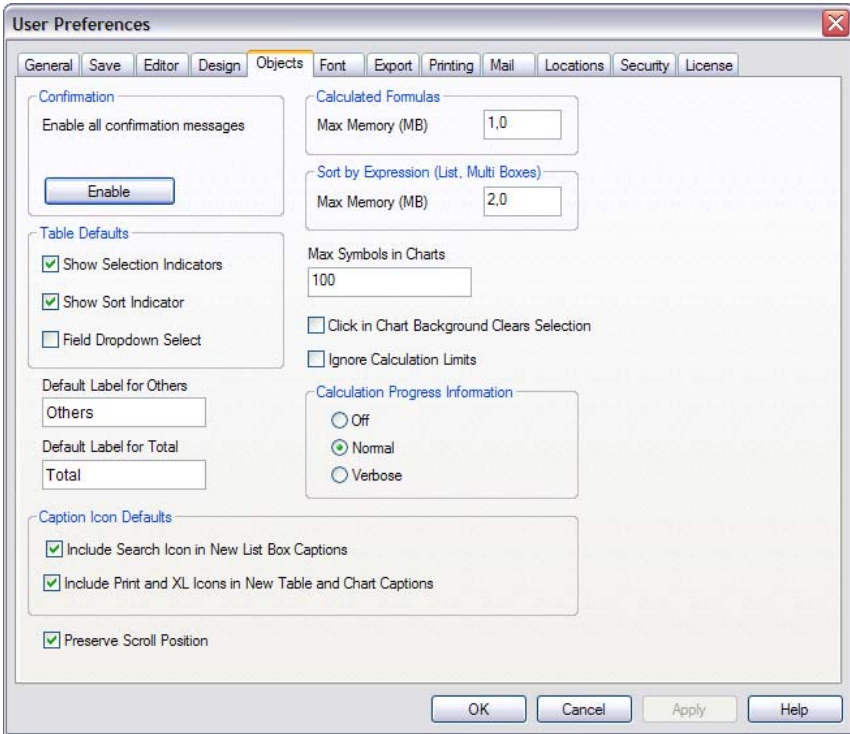


Figure 16. The Objects page in the User Preferences dialog

In the **Objects** page default settings for sheet objects are set.

Confirmation

When QlikView is first installed on a computer a number of warning dialogs are enabled. These will prompt you for confirmation before performing certain actions, such as deleting sheets and sheet objects or sending mail. In each of these warning dialogs there is a check box marked **Do not show this message again**. By marking this check box that particular warning dialog will be suspended for the future. If you want to turn on all previously disabled warning dialogs press the **Enable** button in this group.

Table Defaults

In the **Table Defaults** group you can set defaults for the column icons you want to have in new tables.

Show Selection Indicators

Mark this check box if you wish to have column selection indicators (beacons) turned on as default for new table boxes, pivot tables and straight tables.

Show Sort Indicator

Mark this check box if you want an icon indicating the primary sort column as default for new table boxes and straight tables.

Field Drop-down Select

Mark this check box if you wish to have drop-down selection icons as default in field columns for new table boxes, pivot tables and straight tables.

Default Label for Others

Here you can specify a default label for Others in bar charts and pie charts.

Default Label for Total

Here you can specify a default label for Totals in bar charts, pivot tables and straight tables.

Calculated Formulas - Max Memory (MB)

Here you may specify the maximum memory allocation for the evaluation of calculated formulas. The default is 1 Mbytes. This setting is by default overridden by the **Ignore Calculation Limits** setting.

Sort by Expression (List-, Multi Boxes) - Max Memory (MB)

Here you can specify the maximum memory allocation for sorting by expression. The default is 2.0 MB.

Max Symbols in Chart

Here you may specify an upper limit of the data points to be shown with symbols in line and combo charts. The default value is 100. When the total number of data points exceeds this limit, symbols will be turned off. This feature is only useful in line and combo charts and for expressions with both line and symbols turned on.

Click in Chart Background Clears Selection

If this box is checked, clicking in the background of a chart's plot area will cause all selections in the chart's dimension fields to be cleared.

Ignore Calculation Limits

Earlier versions of QlikView have largely relied on manual settings for limits on how much memory charts and other calculated sheet objects may use. With generally increasing memory sizes and

improved memory management algorithms in QlikView, the old hard controls have become largely obsolete. When this check box is selected (default) the old memory settings will be disregarded and automatic size control, based on available memory, will take effect. When this setting is used the **Memory...** buttons in the various sheet object property dialogs will be disabled. The same applies to the memory limit controls on this page. It is recommended for best performance to leave this check box marked.

Calculation Progress Information

In the **Calculation Progress Information** group you can decide how much progress information should be shown when sheet objects require more than one second to complete calculation.

Off

No progress information is displayed.

Normal

A progress bar is displayed.

Verbose

A progress bar and additional text information is displayed.

Caption Icon Defaults

In this group it is possible to set defaults for selected caption icons.

Include Search Icon in New List Box Captions

When this check box is marked, all new list boxes will have the **Search** caption icon turned on at creation. This is recommended for better usability and especially if the document is to be published for the QlikView AJAX ZFC client.

Include Print and XL Icons in New Table and Chart Captions

When this check box is marked, all new tables and charts will have the **Print** and **Send to Excel** caption icons turned on at creation. This is recommended for better usability and especially if the document is to be published for the QlikView AJAX ZFC client.

Preserve Scroll Position

Mark this check box and the corresponding box on the **Layout** page of a sheet object to enable QlikView to try to preserve the scroll position after a selection in table objects and charts with an x-axis scroll bar.

10.6 Font

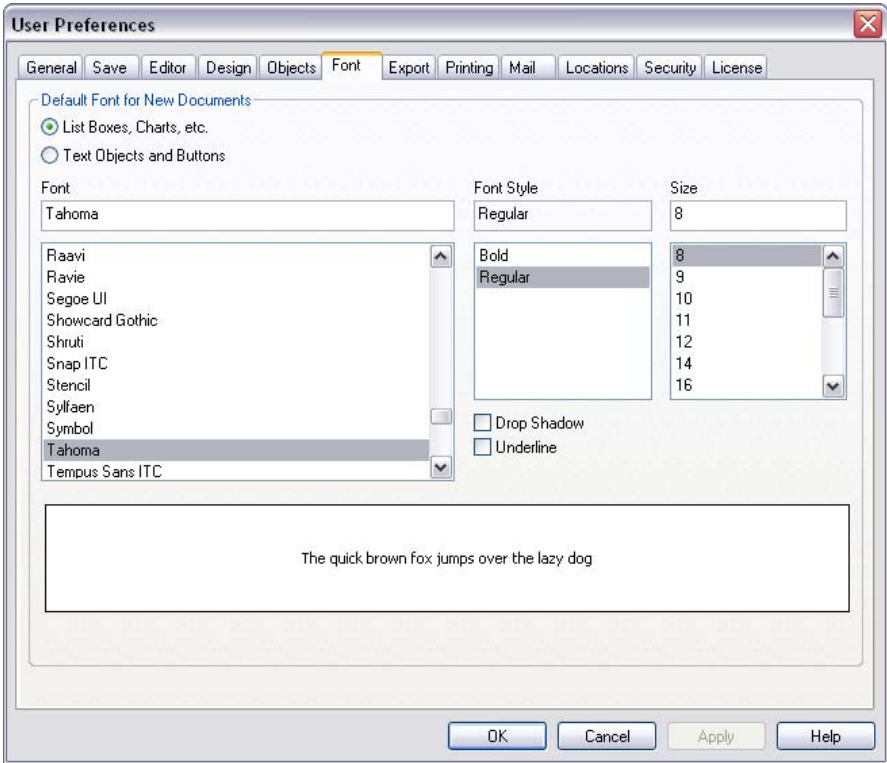


Figure 17. The Font page in the User Preferences dialog

In the **Font** page, you can change one or both of the default fonts for new documents. The first default font is used for most objects, including list boxes and charts. The second default font is used for buttons and text boxes, which are objects that usually need a larger font.

Drop Shadow

If this check box is marked a drop shadow will be added to the text.

Underline

If this check box is marked the text will be underlined.

10.7 Export

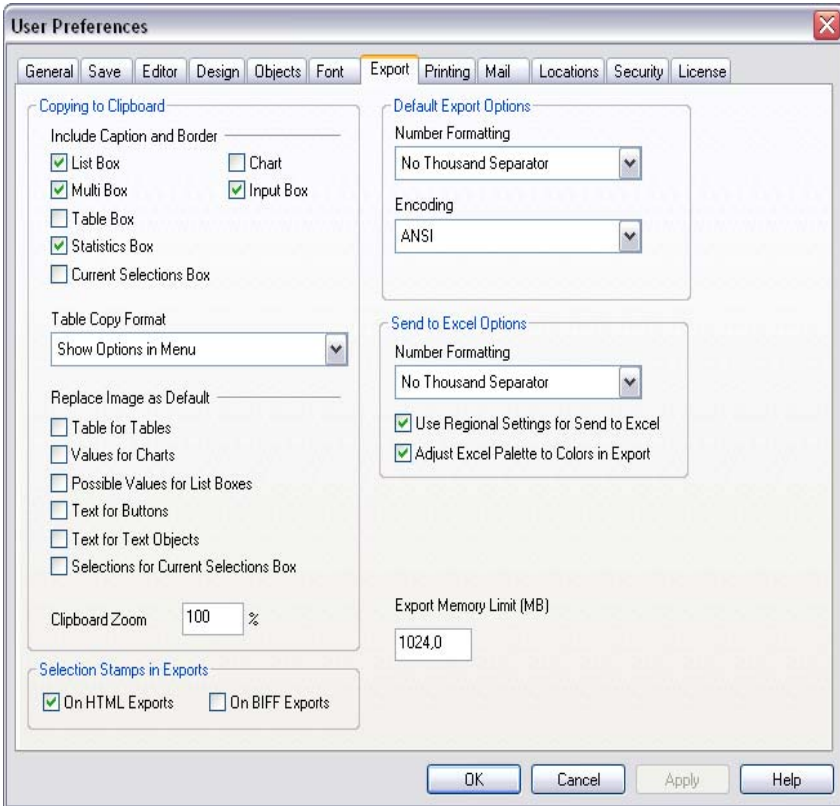


Figure 18. The Export page in the User Preferences dialog

In the **Export** page default settings are made for export and copying to clipboard.

Copying to Clipboard

The **Copying to Clipboard** group contains the preferences for copying sheet objects to the clipboard.

Include Caption and Border

The **Include Caption and Border** group contains the preferences for whether the sheet object caption and border should be included when sheet objects are copied to the clipboard as images. Separate settings are available for list box, statistics box, multi box, table box, input box, current

selections box and chart (including pivot table and straight table).

Table Copy Format

The **Table Copy Format** group contains a drop-down list indicating the preferences for copying tables (straight tables, pivot tables and table boxes) to the clipboard.

Full Table

If this option is selected, the full table (including header and sums) will be copied to the clipboard. Furthermore, an HTML-formatted version of the table will be added to the clipboard for applications that can read it.

Data Area Only

If this option is selected, only the data area of the table will be copied to the clipboard. No HTML version is generated.

Show Options in Menu

If this option is selected, selecting the **Copy Table to Clipboard** command in the table object menu will open a cascading menu containing the commands **Full Table** and **Data Area Only**.

Replace Image as Default

The **Replace Image as Default** group contains the preferences for what should be put on the clipboard to external applications when using the menu commands **Cut** and **Copy** in the **Edit** menu (CTRL+X and CTRL+C). Normally the image of the sheet object is put on the clipboard, but for certain types of sheet objects other options are available.

Table for Tables

If this check box is marked, tables (table boxes, straight tables and pivot tables) will be copied in table format instead of as an image when the **Copy** and **Cut** commands are used.

Values for Charts

If this check box is marked, charts will be copied as underlying values instead of as an image when the **Copy** and **Cut** commands are used.

Possible Values for List Boxes

If this check box is marked, list boxes will be copied as possible values instead of as an image when the **Copy** and **Cut** commands are used.

Text for Buttons

If this check box is marked, buttons will be copied as text instead of as an image when the **Copy** and **Cut** commands are used.

Text for Text Objects

If this check box is marked, text objects will be copied as text instead of as an image when the **Copy** and **Cut** commands are used.

Selections for Current Selection Boxes

If this check box is marked, current selection boxes will be copied as a text selection stamp instead of as an image when the **Copy** and **Cut** commands are used.

Clipboard Zoom

Here you can specify the default zoom factor for copying a sheet object to the clipboard as an image. This setting is independent of the current zoom factor of the sheet. The default zoom factor is 100%. A bigger zoom factor will produce an image of higher quality for scaling, but the object will also be larger on disc.

Selection Stamps in Exports

In the **Selection Stamps in Exports** group you may state whether or not the selection stamps should be included when exporting to certain file types.

On HTML Exports

Mark this check box if selection stamps are to be included when exporting to HTML files.

On BIFF Exports

Mark this check box if selection stamps are to be included when exporting to BIFF (Excel) files.

Default Export Options

In the **Default Export Options** group you may set defaults for export formatting.

Number Formatting

Some other programs may have difficulties to handle numbers with number format correctly. QlikView offers three options for number formatting of numeric data to be exported to files or to the clipboard. This setting will affect the result of all **Export** and **Copy to Clipboard** commands on the menus and of all data transferred to the clipboard in **Copy Mode**. It will also be the default setting for new

export buttons. The setting can however be individually set for each export button (see page 205 in Book II).

Full Formatting

Instructs QlikView to export numeric data with its full number format, just as it is shown in the sheet objects of the document.

No Thousand Separator

Removes any thousand separator from numeric data. This option is recommended if data is to be exported to MS Excel.

No Formatting

Removes all number formatting from the data and exports the raw numbers. The decimal separator will be as defined in the system settings (**Control Panel**).

Encoding

Sets the default character set for export in new documents. You may choose between **ANSI**, **Unicode** and **UTF-8**.

Send to Excel Options

In the **Send to Excel Options** group you may set defaults for formatting used by the **Send to Excel** menu command.

Number Formatting

QlikView offers three options for number formatting of numeric data to be sent to Excel.

Full Formatting

Instructs QlikView to export numeric data with its full number format, just as it is shown in the sheet objects of the document.

No Thousand Separator

Removes any thousand separator from numeric data. This option is recommended if you experience problems with the data sent to Excel.

No Formatting

Removes all number formatting from the data and exports the raw numbers. The decimal separator will be as defined in the system settings (**Control Panel**).

Use Regional Settings for Send to Excel

As long as this check box is selected the regional settings of the operating system will be used for decimal separator

when exporting tables via the **Send to Excel** command. If unchecked, decimal point will always be used. This may be necessary when using certain national versions of Microsoft Excel.

Adjust Excel Palette to Colors in Export

If this check box is marked, the color palette of QlikView replaces that of Excel. Thus the colors of that Excel document will differ from those in other Excel documents, but will be closer in hue to the colors of the original document.

If unchecked, the standard colors of the Excel palette will replace the colors chosen in QlikView. The original colors will be replaced by those that correspond the best to the colors in the standard palette.

Export Memory Limit (MB)

Sets the maximum amount of RAM that can be used for performing an export from a QlikView sheet object.

10.8 Printing

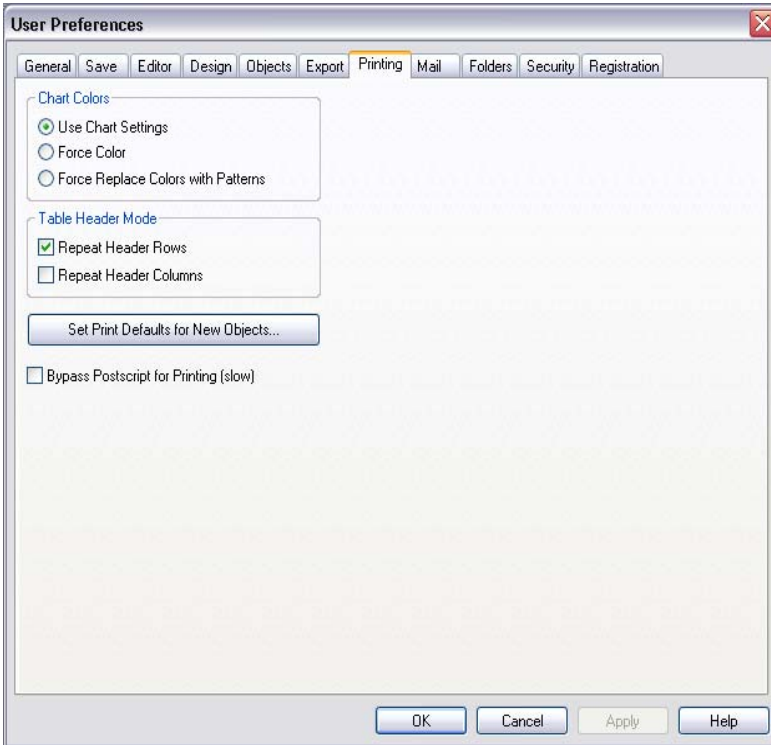


Figure 19. The Printing page in the User Preferences dialog

This group contains the preferences for printing sheet objects.

Chart Colors

In this group you may specify an override of the color print settings of all the image charts in your documents.

Use Chart Settings

When this option is selected, the individual settings for each image chart regarding color or black/white printing will be used.

Force Color

When this option is selected, all image charts will be printed in color, regardless of printer settings on the **Color** page of the **Chart Properties** dialog (see page 70 in Book III).

Force Replace Colors with Patterns

When this option is selected, all image objects will be printed in black/white, regardless of the settings on the **Color** page of the **Chart Properties** dialog (see page 70 in Book III). Color coding will be replaced by pattern coding.

Table Header Mode

When printing tables whose contents expand over several pages, you may want the header rows or columns to be displayed on all the pages. This is when the commands in this group are useful.

Repeat Header Rows

If this check box is selected, the header rows will appear on all the pages of the printed table.

Repeat Header Columns

If this check box is selected, the header columns (in pivot tables) will appear on all the pages of the printed table. If there is not room for at least one data column on the right-hand side of the header columns on one page, this setting will be disregarded.

Set Print Defaults for New Objects

Opens a dialog where default margins and print orientation can be set.

Bypass Postscript for printing (slow)

Due to the interaction between Microsoft graphics libraries and certain printer Postscript drivers, printouts from **Print Sheet** may sometimes be less sharp than expected. This effect can be avoided by marking this check box. Note however, that printing times may become considerably longer (up to several minutes).

10.9 Mail

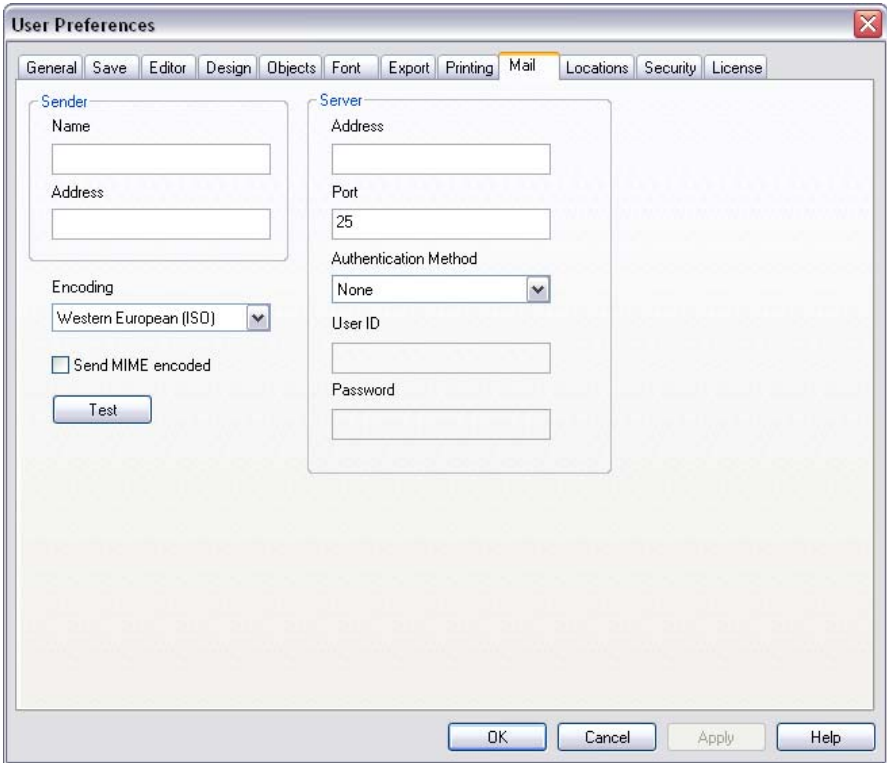


Figure 20. The Mail page in the User Preferences dialog

In this page you can make settings for sending mails from QlikView. In order to the mail functionality you must have access to an SMTP server.

Sender

In this group you specify the name and e-mail address that will appear as sender on the e-mails sent by QlikView.

Name

The name that should appear as sender for outgoing e-mails.

Address

The e-mail address that should appear as sender for outgoing e-mails.

Encoding

Select the character code page to be used for sent mail.

Send MIME encoded

Mark this check box if mails are to be MIME encoded.

Server

In this group you specify the SMTP server to be used for outgoing mails from QlikView.

Address

The address (URL or IP number) of the SMTP server to be used.

Port

The port used by the SMTP server.

Authentication Method

Select an authentication method from the list, if authentication is required by the SMTP server.

User ID

The user ID to be used for authentication.

Password

The password to be used for authentication.

10.10 Locations

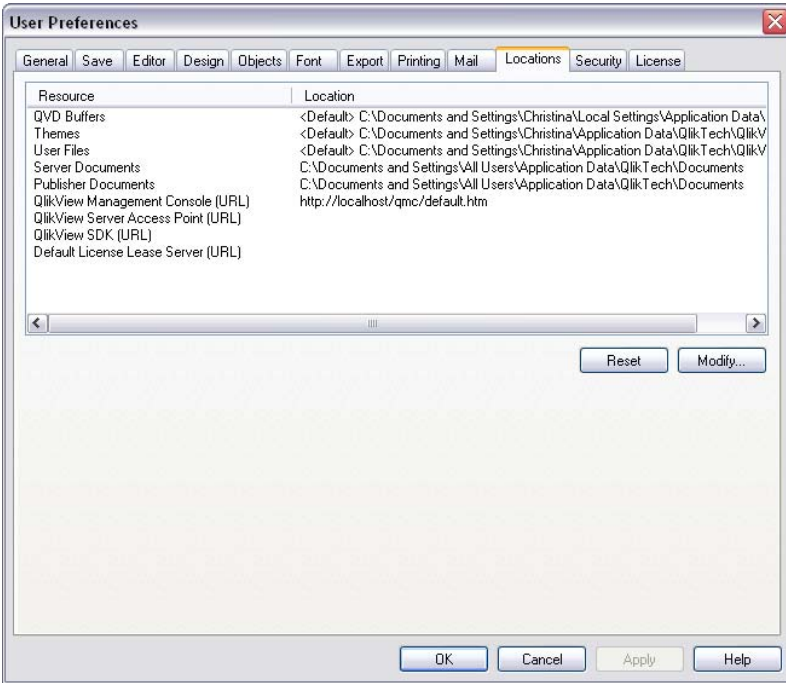


Figure 21. The Locations page of the User Preferences dialog

The **Locations** page is used for setting default folder locations for certain files created when working with QlikView. It can also be used to define shortcuts to document file folders for QlikView Server and QlikView Publisher. Finally you may specify URLs to the control panels of QlikView Server, QlikView Publisher and QlikView AccessPoint. The page contains a list of resource locations that can be modified.

Resource

The following resource locations can be modified:

QVD Buffers

The default storage location for QVD buffers generated via the **buffered** prefix to **load** and **select** statements in the script.

Themes

The default storage location for user defined layout themes.

User Files

The default root folder location where the folders storing user bookmarks, user reports and user alerts are stored. Note that by changing this location without moving the folder contents you will lose your existing user bookmarks, user reports and user alerts.

Server Documents

Here you may specify the location of your QlikView Server document folder, when applicable. If you do, you will find a shortcut icon to that folder in the **Open** and **Save As** dialogs.

Publisher Documents

Here you may specify the location of your QlikView Publisher source document folder, when applicable. If you do, you will find a shortcut icon to that folder in the **Open** and **Save As** dialogs.

QlikView Management Console (URL)

Here you may specify a URL pointing at your QlikView Management Console, QMC, or QlikView Enterprise Management Console, QEMC, when applicable. If you do, you will find a shortcut icon to the control panel in the **Open** and **Save As** dialogs.

QlikView Server Access Point (URL)

Here you may specify a URL pointing at your QlikView AccessPoint, when applicable. If you do, you will find a shortcut icon to that folder in the **Open** and **Save As** dialogs.

QlikView SDK (URL)

Here you may specify a URL pointing to the QlikView SDK, when applicable.

QlikView License Lease Server (URL)

Here you may specify a URL pointing to your QlikView License Lease Server, when applicable.

Reset

Resets the location of the selected resource to the QlikView default. The path shown in the list will be preceded by the text **<default>**.

Modify...

When a folder resource is modified this button opens the **Browse for Folder** dialog where you may browse to the location you prefer

for the selected folder. When a URL resource is modified this button opens a dialog where you can enter a URL.

10.11 Security

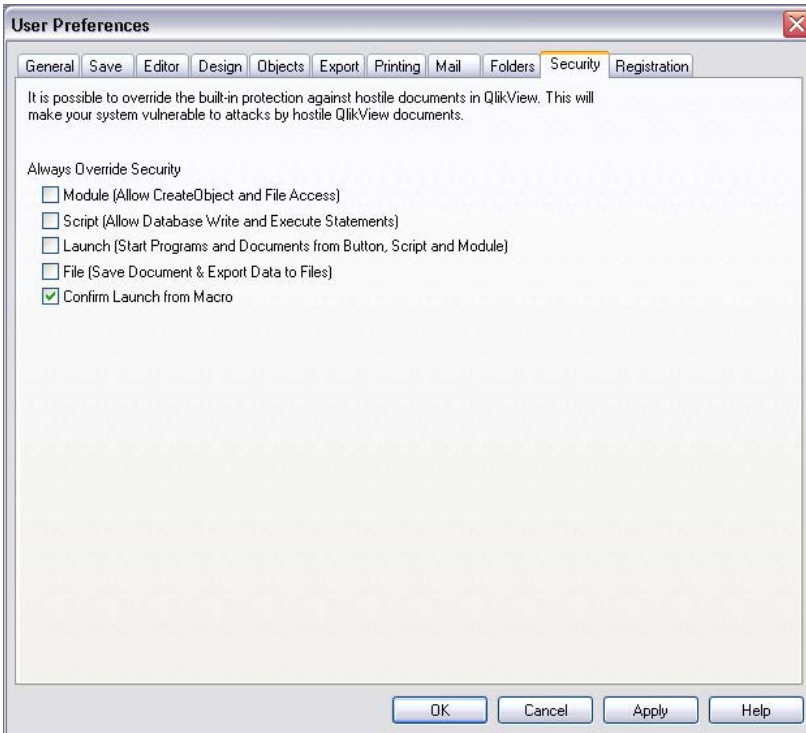


Figure 22. The Security page in the User Preferences dialog.

In this page you can choose to override one or more parts of the QlikView security measures against hostile macros and scripts embedded in the QlikView document. Use these options with caution and only when working with well-known documents.

Module (Allow CreateObject and file access)

By marking this box you turn off QlikView checks for macros containing CreateObject calls or accessing external files.

Script (Allow database write and execute statements)

By marking this box you turn off QlikView checks for scripts containing the **execute** command and **mode is write** qualifier in **select** statements.

Launch (Start programs and documents from button, script, module)

By marking this box you turn off QlikView checks for starting external programs and documents from QlikView script, module or buttons.

File (Save doc and export data to file)

By marking this box you turn off QlikView checks for saving or exporting to files with suspect file extensions.

Confirm Launch from Macro

By keeping this check box marked, QlikView will have you confirm that you permit a macro to launch another application.

10.12 License

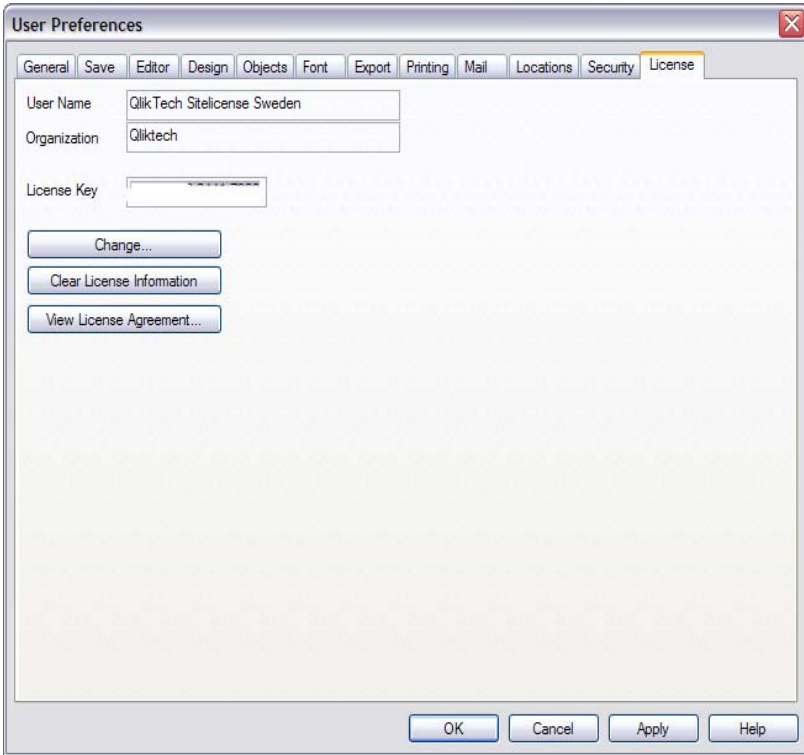


Figure 23. The License page in the User Preferences dialog.

This page is used only when you need to change the serial number of your QlikView license. Changes take effect only after restarting QlikView.

Change...

By clicking this button, you will have the possibility to enter a new serial number and a new control number for your license. Changes take effect only after restarting QlikView.

Clear License Information

By clicking this button, your license number will be erased the next time the application is started.

View License Agreement

Clicking this button will show the license agreement.

11 CALCULATION MEMORY SETTINGS

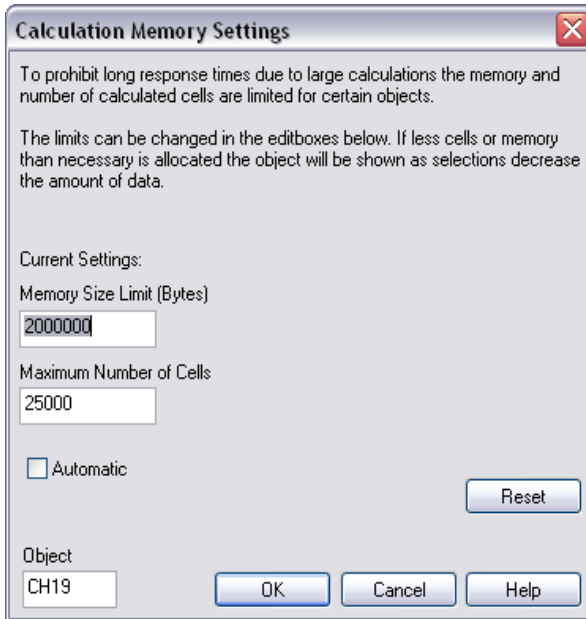


Figure 24. The Calculation Memory Settings dialog.

Charts and table boxes in QlikView are so called calculated sheet objects. This means that they do not only show field values, but rather the result of a sometimes complex calculation based upon field values.

In the case of the table box, the calculation involves finding all possible combinations between field values in the selected fields.

In the case of charts (including pivot tables and straight tables) the calculation involves two steps: first all possible combinations of field values in the dimensions must be found. Then the expressions must be evaluated over these combinations.

All calculated objects require some extra memory for the calculation. The amount of memory needed naturally depends on the complexity of the calculation. QlikView imposes very few hard limitations on the complexity of your calculated objects. However, if you decide to make a ten-dimensional pivot table over several million

records, a substantial amount of RAM will be required. Windows will, if it runs out of physical RAM, start swapping memory to disk. This will slow down the computer considerably.

If you make a chart or table box large enough, the program may actually appear to hang while trying to calculate the object. In order to prevent this from happening, QlikView may limit the amount of memory given to any calculated sheet object. When more memory is required, QlikView will display the error message "Out of object memory" or "Too many Cells".

In most cases it is wiser to leave the setting of limits to QlikView. QlikView will do so taking into account the available RAM and other factors. Provided that the **Ignore Calculation Limits** setting has been selected in the **Objects** page of the **User Preferences** dialog, all memory limits will be handled automatically. Only if this setting is deselected will the **Calculation Memory Settings** dialog become available under **Chart Properties** and **Table Box Properties**. With this dialog you may set hard custom limits to specific sheet objects.

There are two separate limits available. First, you may run out of memory for the calculation and second, you may generate a table too large in number of cells. Please note that the cells require memory also after the calculation, as long as the object is displayed on the active sheet.

You may choose to increase **Memory Size Limit** or **Maximum Number of Cells** until the calculation can be performed. It is also possible to leave the object with insufficient memory. In many cases the calculation will be made possible as the data set is narrowed down by selections in the data.

As long as a calculated object cannot be properly calculated, it will show a message saying "Out of object memory" or "Too many Cells".

You may access the **Calculation Memory Settings** dialog for a calculated sheet object via the **Memory...** button in the appropriate **Properties** dialog.

Memory Size Limit (Bytes)

Here you may change the amount of allocated memory.

Maximum Number of Cells

Here you may change the maximum number of cells.

Automatic

By marking this check box you turn on automatic limits for the specific sheet object even if the **Ignore Calculation Limits** setting has been deselected in the **Objects** page of the **User Preferences** dialog.

Reset

Resets all values to the defaults defined on the **Objects** page in the **User Preferences** dialog (see page 107).

Object

Here you can see the sheet object ID of the current chart or table box.

OK

Accepts changes and closes the dialog.

Cancel

Closes the dialog without saving the changes.

Help

Invokes help.



12 LOGIC AND SELECTIONS

12.1 Single Field Value Selection

In order to select a value from a field, simply click the value. Once selected, the cell turns green to indicate its new state, i.e. selected. The selection may affect the states of a large number of values in the other sheet objects. White cells represent optional field values, whereas gray cells represent field values excluded by selections. When a selection is made, natural inner joins are dynamically created between all the associated tables.

Selections can be made not only in list boxes, but also in statistics boxes, multi boxes, table boxes and charts. See page 138 for more information about selections in objects.

To cancel a previously made selection, click it or choose **Clear** from the **Object** menu, or choose one of the **Clear** commands in the **Selections** menu. When an excluded value is clicked, selections in conflict with this value are canceled, and the value clicked becomes selected.

Color Codes

The state of a field value is shown by the color of its cell. The following color scheme is used:

Select	Green
Optional	White
Alternative	White
Locked	Blue
Excluded	Gray
Forced excluded	Red

When the **option** is on, i.e. checked (**Show Alternatives** page 102 in Book II) QlikView displays non-selected cells in the corresponding list box as alternative values (white), unless they are excluded by selections in other list boxes. They are however *logically excluded*, and not included in calculations made on possible (optional and selected) values. When the **Show Alternatives** option is off, i.e. not marked, QlikView displays non-selected cells as excluded (gray).

12.2 Multiple Selection within a Field

Multiple selections within a field (from one list box) can be made in a number of ways:

- Position the cursor in the list box, then drag it over a number of field values while pressing the mouse button.
- Click the first field value to be selected, then CTRL-click each additional selection.
- Click the top item to be selected, then SHIFT-click the bottom item to be selected. This way all the items in between will be selected. However, if the sort mechanism is on, the first selection may cause the order to change, which makes it difficult to make a correct second selection.
- Confirm a text search (see below) by hitting the ENTER key. This will result in all matching field values being selected. By keeping the CTRL key depressed while pressing ENTER the selections from the text search will be added to previous selections.

To deselect a previously made additional selection, CTRL-click it.

If you have made a multiple selection in one list box, and make a new selection from the available optional values in another list box, some of the selected values in the first list box may get excluded. However, when the selection in the second list box is cancelled, your previous selections will, by default, be recovered.

A multiple selection can be interpreted in two different ways, either as a logical **or** or as a logical **and**. Default is logical **or**, i.e. QlikView will find a solution that is associated to one or more of the selected field values. For logical **and**, see page 137.

12.3 Search

Selections can be made through text search as well. To enter a search string, click on the list box header, then simply type the search string. The string is not case sensitive. The search string will appear in the pop-up search box. As a result, QlikView will display all the values of the selected field that fulfill the criteria of the search string. When you press ENTER, or click on one of the cells in the result, the value(s) become(s) selected. By keeping the CTRL key depressed while pressing ENTER the selections from the text search will be added to previous selections.



The search box will close automatically when you hit ENTER, ESC or click in the layout. You may also close it by clicking the **x** icon in the search box. The search box is sizable and will retain its size when it is opened again.

If several list boxes are active (SHIFT-click on their headers to make them active), they are all included in the text search. Pressing ENTER to select the resulting values is however not possible until there are optional values in only one of the active list boxes.

If a selection has already been made, there are two ways in which a search can be interpreted: either you search only among the optional values, or you search among all values, i.e. you include the excluded values in the search. To set the search mode to work in, you can either select or deselect **Include Excluded Values in Search** in the **User Preferences** dialog. This mode can also be set on certain individual sheet objects.

If the logical **and** option is set for a field, it may not be possible to select multiple values found.

Text Search

The simplest way of searching is text search. QlikView will search for field values matching a text string that you type. The search string typically contains wildcard characters. * denotes any number of arbitrary characters whereas ? denotes a single arbitrary character. When you start typing text it will by default appear between two wildcard characters, i.e. you will search for any field value containing the search text. Wild cards may appear several times in the search string, regardless of their location.

* Zero or more characters.

? Any single character.

The ENTER key can be pressed to select the values found, and the ESC key can be pressed to cancel the operation.

Examples:

a* will find all values that begin with the letter a.

b will find all values that contain the letter b.

Fuzzy Search

If you start your text search with a tilde sign ~ the text search window will be opened in fuzzy search mode. The search window will contain the tilde with the cursor placed after it. As you type all values will be sorted by the degree of resemblance to the search string with the best matches at the top of the list. If you hit ENTER the first value in the list will be selected.

Numeric Search

Selections can be made through numeric search as well. This is very similar to text search. The only difference is that the search string must begin with one of the relational operators ">", ">=", "<" or "<=".

Examples:

- >900 will find all values greater than 900.
- <=900 will find all values less than or equal to 900.
- >900<1000 will find all values greater than 900 and less than 1000.
- <900>1000 will find all values less than 900 or greater than 1000.

Advanced Search

For complex search expressions you can use the **Advanced Search** dialog (see below), which can be invoked by the keyboard shortcut CTRL+SHIFT+F. If you start your text search with an equal sign =, you may enter an advanced search expression involving search criteria for associated fields and full boolean logic. After the equal sign you may type any valid QlikView layout expression (see chapter 14 in Book III). The expression will be evaluated for each field value in the search field. All values for which the search expression returns a non-zero value will be selected.

Examples:

=*MyField* like 'A*' or *MyField* like '*Z'

if e.g. invoked from a list box containing the field *MyField*, the search would return all field values either starting with the letter A or ending with the letter Z.

=sum(*Sales*)>sum(*Budget*)

if e.g. invoked from a list box containing the field *Salesman*, the search would return all salesmen with an associated sales value larger than their associated *Budget*.

12.4 The Advanced Search Dialog

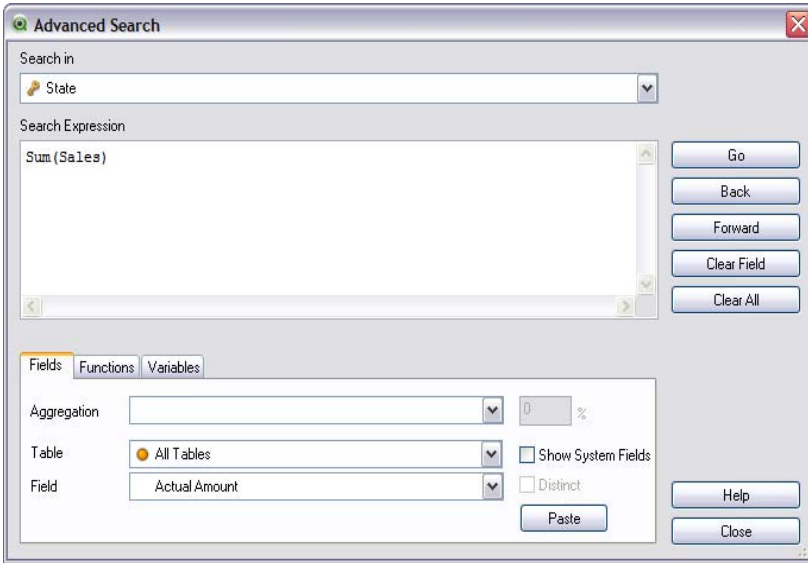


Figure 25. The Advanced Search dialog

This dialog is provided as a convenient way of formulating complex search queries to be applied on QlikView fields. Unlike the standard search box that appears when starting a search by typing when a list box is active, the result of a search is not interactively visible in the layout until you submit the search by pressing the **Go** button. The dialog can be kept open while you work in the QlikView layout. The dialog is fully sizeable in order to facilitate editing of large and complex expressions.

Search in

The field in which the search will be made. When you enter the dialog it will be set to the field of the active list box. You may change search field by using the drop-down at any time.

Search Expression

This is where you type the search expression. The same rules apply as when using the normal search box (see previous section).

Go

Applies the search to the search field.

Back

QlikView remembers the last 100 selections. By clicking this button, you go one step back in the list of selections.

Forward

By clicking this button, you go one step forward in the list of selections (equivalent of cancelling the last **Back** command). This is only possible if the **Back** command has been used immediately before.

Clear Field

Clears selections in the current search field.

Clear All

Clears all selections in the document.

Help

Opens the **Help** dialog for advanced search.

Close

Closes the dialog box.

In the bottom pane of the dialog you will find three tabs which can help you when building advanced search expressions.

Fields

On the **Fields** tab you will find controls for pasting syntax relating to QlikView field data.

Aggregation

In this drop-down you can choose from the statistical aggregation functions available in the QlikView layout.

Table

In this drop-down you may select a specific input table from which to pick fields in order to make navigation in the **Field** drop-down easier.

Field

In this drop-down you can choose from all the available fields.

Show System Fields

If this check box is marked, the list containing the fields of the document includes the system fields.

Distinct

The statistical functions are by default calculated on the number of occurrences in the original table. Sometimes, however, you do not want to calculate duplicates. If this is the case, mark this check box before pasting the function.

Paste

Pastes the selected function or just the field into the Search Expression edit box. A percentage can be given when using the fractile function.

Functions

On the **Functions** tab you will find controls for pasting syntax relating to QlikView general functions.

Function Category

In the drop-down you may select a category of functions in order to make navigation in the **Function Name** drop-down easier.

Function Name

In the drop-down menu you may select a function for pasting into the expression out of all functions available in the QlikView layout. The list can be reduced to show only functions belonging to a certain category by means of a selection in the Function Category drop-down above.

Paste

Pastes the selected function name into the Search Expression edit box.

At the bottom of the tab there is a pane showing the argument syntax of the function selected in the **Function Name** drop-down.

Variables

On the **Variables** tab you will find controls for pasting syntax relating to QlikView variables.

Variables

In the drop-down you will find all currently defined variables in the document.

Paste

Pastes the selected function into the **Search Expression** edit box.

Show System Variables

If this check box is marked, the list in the **Variables** drop-down will include the system variables.

At the bottom of the tab there is a pane showing the current value of any variable selected in the **Variables** drop-down.

12.5 Moving the Selection

The current selection in an active list box can be moved by means of keyboard keys:



Moves the current selection one step downwards in the list in the current sort order. If more than one value is selected, each selection

is stepped downwards. When the last value in the list box is passed, the selection is wrapped back to the first value. If no selection is made in the list box, the list box will be scrolled one cell downwards.



Moves the current selection one step upwards in the list in the current sort order. If more than one value is selected, each selection is stepped upwards. When the first value in the list box is passed, the selection is wrapped down to the last value. If no selection is made in the list box, the list box will be scrolled one cell upwards.

PgDn

Moves the current selection downwards in the list in the current sort order by the same number of cells as the maximum distance between the first and last selected values in a multiple selection. When the last set of values in the list box is passed, the selection is wrapped back to the first set of values. If no selection is made in the list box, the list box will be scrolled one page downwards.

PgUp

Moves the current selection upwards in the list in the current sort order by the same number of cells as the maximum distance between the first and last selected values in a multiple selection. When the first set of values in the list box is passed, the selection is wrapped down to the last set of values. If no selection is made in the list box, the list box will be scrolled one page upwards.

Home

Moves the current set of selections upwards to the top of the list.

End

Moves the current set of selections downwards to the end of the list.

Example:

Assume you have the following list box (E and F selected):



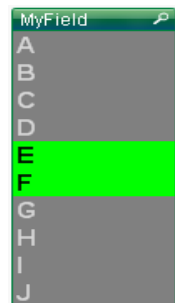
F and G will be selected.



D and E will be selected.

PgDn

G and H will be selected.



PgUp

C and D will be selected.

Home

A and B will be selected.

End

I and J will be selected.

12.6 Logical *And* Mode

A multiple selection within a list box is usually interpreted as logical *or*, meaning that data entries associated with any of the selected values will be shown. However, under certain circumstances it can be set to be interpreted as *and*, implying that QlikView must find the solutions that are associated to all the selected field values, not just to any of them. The list box is then set to *and* mode.

The mode for the field is set in the list box dialog (**Object** menu, **Properties, General, And mode**).

A value in a list box in *and* mode will be marked with an ampersand "&" when selected.

And-Tables

A field cannot always be set to logical *and* mode. The reason for this is that the *and* alternative is logically meaningful only if the concerned field is linked to only one other field. The following criteria must be fulfilled:

- the field must only exist in one logical table,
- the field must be the second column, and
- the table must not contain any duplicate records and as of version 7.0
- the field must be distinct.

Forced Exclusion / Not Selection

Closely related to the *and* selection is the forced exclusion, or *not* selection. Here it is possible to explicitly exclude a field value, i.e. the solutions found by QlikView may not be associated with the excluded value.

The *not* selection is made by clicking a cell and *keeping the mouse button depressed* until the cell turns red. CTRL-clicking in this manner is equivalent to an exclusion that is an additional requirement to the previous selections/exclusions.

The *not* selection can only be made on a field that is in *and* mode.

12.7 Locking Selected Field Values

After selecting an excluded field value, previous selections in conflict with the new choice are canceled. To prevent a previous selection from getting canceled this way, you can lock all the selections in a list box by opening its **Object** menu and choosing the **Lock** command.

To unlock a previously set lock, open the list box **Object** menu, and select **Unlock**. General locks and unlocks of all the selections can be made from the **Selections** menu.

When you try to select a value which is incompatible with a locked selection in another field, the selection will fail.

By means of the **Override Locked Field** setting for list boxes, multi boxes and slider objects it is possible to override a locked selection in a field from a specific sheet object. The field will still be locked for logical changes stemming from selections in other fields. This option is on by default for slider objects.

12.8 Selections in Other Objects

Selections in field data can be made directly in most of QlikView objects by means of clicking or painting with the mouse. The following section describes the possibilities for selections in the different types of objects.

Statistics Boxes

In statistics boxes you can click on some of the statistical quantities, e.g. **Min**, **Max** and **Median**, and the corresponding value will be selected. The selection is not marked in the statistics box but only in other boxes.

Multi Boxes

A row in the Multi Box represents a field. Clicking on the small arrow displays a list of values that belong to the field. Selections and searches can be made in this list just as in a list box.

Table Boxes

Selections can be made by clicking in any cell or by painting over an area covering one or more rows and one or more columns.

If the option **Drop-down Select** is active, an arrow is displayed in the column heading. Clicking on the arrow displays a list of the values that belong to this field. Selections and searches can be made in this list just as in a list box.

Slider/Calendar Objects

In sliders, where a single field is the basis, it is possible to select a value by adjusting the thumb tack to the desired position. If the slider is accordingly configured, the size of the thumb tack can be changed with a mouse click. This way it is possible to select several values.

Clicking on the small calendar symbol in a calendar object opens the calendar. There you can select a date or a whole period with the mouse depending on the configuration of the calendar object, and this selection is transferred to the underlying field. Using CTRL+CLICK you can select several periods even if they are in different months or years.

Bar, Line, Combo, Radar, Grid and Scatter Charts

Selections can be made inside the plot area by clicking on a single data point or painting over several data points. When painting, the area covered is shown with a green raster. The selection will be made for the dimension values used to calculate the selected data point(s).

Selections can be made by clicking or painting in the chart legend (except when the legend is indicating chart expressions rather than dimension values).

Selections can be made by clicking or painting over the dimension axes and its labels (except scatter charts). The corresponding field values will be selected.

Selections can be made by painting over the expression axes and its labels. The field values which generate data points in the indicated result area will be selected.

When painting selections in line charts and bar charts with more than one dimension the behavior of the QlikView selection logic differs slightly from other that of other charts in order to better reflect the expectations of the user. Selections in these types of charts do not affect both dimensions at the same time. In line charts selections will be primarily done in the second dimension. This means that painting over a line will select the entire line over all x-axis dimension values. In bar charts the opposite applies. Selections primarily apply to the first dimension. This means e.g. that clicking one bar segment will result in a selection of that segment's x-axis dimension value but leave all stack or cluster segments possible. When selections have narrowed down the primary selection dimension to one single value, the old selection logic will apply again, making selections prevail also in the secondary selection dimension. In combo charts selections always affect all dimensions. If you find the above explanation incomprehensible, do not worry! The logic probably works like you would like it to do anyway!

Pie Charts

Selections can be made inside the plot area by clicking on a single pie slice or painting over several slices. When painting, the area covered is shown with a green raster. The selection will be made for the dimension values used to calculate the selected data point(s).

Selections can be made by clicking or painting in the chart legend.

Block Charts

In block charts individual blocks can be selected. With selections the drill-down functionality comes into play. Selecting the first block refers to the first dimension, selecting a second block within the first refers to the second dimension etc.

It is also possible to select several blocks by painting an area. This selected area is marked green until the mouse button is released. Such a selection refers back to the value or values of the first dimension. Based on these values the corresponding blocks are calculated. If a selection crosses the block borders of several values that belong to the first dimension, it will affect all related values belonging to the second and third dimensions as well, not only those in the selected area.

Gauge charts

Selections cannot be made in gauge charts.

Straight Tables

Selections can be made in the dimension columns by clicking on a cell or painting over several cells. The selected area is marked green until the mouse button is released.

If **Drop-down Select** is activated in a column that represents a dimension, a small arrow will be displayed in the column header. Clicking on the arrow will display a list of all values of the field. Selections and searches can be made in this list.

Selections can be made in the expression columns by clicking in a single cell. The selection will be made for the dimension values used to calculate the selected expression cell.

Pivot Tables

Selections can be made in the dimension columns/rows by clicking on a single cell. The selected cell is marked green until the mouse button is released.

If **Drop-down Select** is activated in a column that represents a dimension, a small arrow will be displayed in the column header. Clicking on the arrow

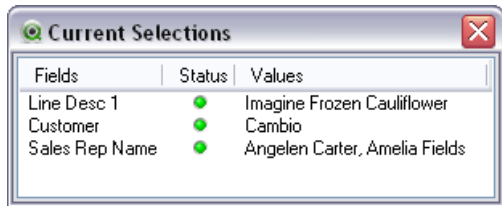
will display a list of all vaules of the field. Selections and searches can be made in this list.

Selections can be made in the expression columns/rows by clicking in a single cell. The selection will be made for the dimension values used to calculate the selected expression cell.

12.9 Current Selections

When many selections have been made in your document, it can sometimes be hard to get an overview of them. This is why the **Current Selections** window has been created.

The **Current Selections** window, which you open by choosing **Current Selections** from the **View** menu or by clicking the **Current Selections** button in the toolbar, contains a list of all the selections made in the document. The names of the fields in which the selections, or searches that led to selections, were made, the status of the selections, and the names of the selected values are listed.



It is possible to highlight one or several fields in the list by clicking them (to highlight several fields, SHIFT- or CTRL-click them). Right-clicking in the **Current Selections** window when one or several fields are highlighted opens a context menu containing the following commands:

Select Excluded

Selects the currently excluded values of the highlighted field. This command is not available when more than one field is highlighted.

Select All

Selects all the values of the highlighted field. This command is not available when more than one field is highlighted.

Clear

Deselects the selected value(s) of the highlighted field(s).

Clear Other Fields

Selects all the possible values of the highlighted field, then clears the selections of all the other fields. This command is not available when more than one field is highlighted.

Lock

Locks the selected value(s) of the highlighted field(s).

Unlock

Unlocks the locked value(s) of the highlighted field(s).

Data

Copies the name(s) of the highlighted field(s) and the name(s) of the selected value(s) in those fields to the clipboard.

If you right-click in the **Current Selections** window without having highlighted a field, the context menu contains the following commands:

Clear All Selections

Deselects all the selected values of the document.

Lock All Selections

Locks all the selected values of the document.

Unlock All Selections

Unlocks all the locked values of the document.

Data

Copies the name(s) of all the selected value(s) of the document, as well as the name(s) of the field(s) in which the selections were made, to the clipboard.

The **Current Selections** window resembles a sheet object in the way that it is sizable and can be kept open while you work with your document. It will be left open when you quit QlikView and reappear the next time you use the program. Just like the sheet objects, it is dynamically updated as soon as new selections are made, thus allowing you to always have an overview of your current selections. For more information about the sheet object **Current Selections Box**, see page 171 in Book II.

The maximum number of distinct selected values to be shown in the **Current Selections** dialog is specified in the **User Preferences** dialog (**General** page). When more values are selected, they will only be displayed as 'x values of y' for the respective field.

Selections made in **and** boxes are preceded by & or !, depending on whether it is a selection or an exclusion.

12.10 Selection Styles

QlikView supports a number of different ways of presenting data and making selections in list boxes and multi boxes. The **QlikView Classic**, **Corner Tag** and **LED** styles all use color coding for indication of selected, possible and excluded values. The **Windows Check boxes** style mimics the standard Windows interface with check box at each value.

When using the selection styles based on color, there are a number of different color schemes available. The basic colors (green for selected, blue for locked etc.) cannot be changed, but variations in tone and intensity are possible.

QlikView Classic is the only style available in the QlikView Java client.

Which style is used can be controlled via settings on the **General** page of **Document Properties** (see page 16 in Book II) and the **General** page of **User Preferences** (see page 91).

This entire manual as well as the **Tutorial** are written for the **QlikView Classic** mode. The differences in the **Windows Check Boxes** mode are described below.

Representation of Logical State

The logical state of data in QlikView is in the color-coded **QlikView Classic** style represented by color coding of background and text in the cells of list boxes and multi boxes. In some other sheet objects and on the sheet tabs so called selection beacons may appear to indicate selections. The & and ! characters may appear to the left of data in list boxes and multi boxes in order to indicate AND and NOT selections.

With the **Windows Check Boxes** style the only color coding remaining is the white and gray background of cells to indicate possible and excluded values in list boxes and multi boxes. However, all cells will also feature an icon to the data, which shows the logical status. These icons also appear in the place of the selection beacons in other parts of the document.

The differences are summarized in the table below:

	QlikView color-coded styles	Windows Check Box Style (icons)
QlikView State		
Possible	White/Black	<input type="checkbox"/>
Excluded	Gray/Light Gray	<input type="checkbox"/>
Selected	Green/Black	<input checked="" type="checkbox"/>
Select Excluded	Gray/Light Gray	<input checked="" type="checkbox"/>
Locked	Blue/Yellow	<input type="checkbox"/>
Locked Excluded	Gray/Light Gray	<input type="checkbox"/>
And Selected	Green/Black with &	& <input checked="" type="checkbox"/>
Not Selected	Red/Black with !	! <input checked="" type="checkbox"/>



Logical Behavior

The logical behavior of the **Windows Check Boxes** style differs in two ways from that of the color-coded styles.

- 1 All selections under the **Windows Check Boxes** style are considered toggle selections, i.e. they work as if you hold down the CTRL key while making selections under the color-coded styles.
- 2 All list boxes are treated as if the **Show Alternatives** option is selected, regardless of the actual setting of this property.

12.11 Loosely Coupled Tables

A loosely coupled table is one where the normal QlikView logic has been disconnected internally. This means that selections in one field do not propagate through to the other fields in the table. This chapter will show some examples of how loosely coupled tables alter the QlikView logic.

A Basic Example

Have a look at the following three table boxes, each one representing a table read into QlikView:

Tab1	
B	A
1	x
2	y
3	z

Tab2	
A	C
x	6
y	7
z	8

Tab3	
C	D
6	a
7	b
8	c

If you select the value 2 in field B the following will happen:

Tab1	
B	A
2	y

Tab2	
A	C
y	7

Tab3	
C	D
7	b

The selection ripples through all tables. Now let us keep this selection but make Tab2 loosely coupled. This means that the logic will be cut between the fields A and C in Tab2. The result will look this:

Tab1	
B	A
2	y

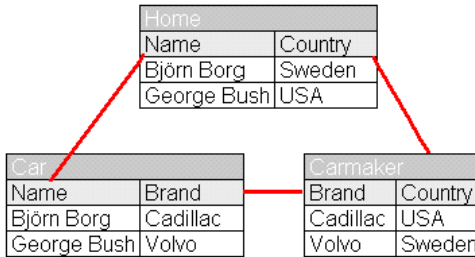
Tab2	
A	C
y	6
y	7
y	8

Tab3	
C	D
6	a
7	b
8	c

Note that Tab2 shown here is a table box and not the table itself. The table box will show all possible combinations between the fields of its columns. Since there is no logic between fields A and C all combinations of their respective possible values are shown.

Avoiding Circular References

The next example shows how loosely coupled tables can be useful for avoiding circular references in the data structure:



This data structure is actually not a very good one since the field name *Country* is used for two different purposes. In one table it indicates where the car owner lives and in another it shows where the car manufacturer resides. With the data in the tables you face an impossible logical situation. Whenever you make a selection, it is possible to follow associations leading through every single cell in all three tables.

You have to decide whether the home country or the car manufacturing country is most important. If you make the *Carmaker* table loosely coupled, the associations from *Cadillac* to *USA* and from *Volvo* to *Sweden* will be broken. By clicking *Sweden* you will associate with *Björn Borg* and *Cadillac*. By clicking *Volvo* you will associate with *George Bush* and *USA*.

Would you prefer to concentrate on car manufacturers it would make sense to make the *Home* table loosely coupled instead.

Another Example

Let us have a look at another common situation where loosely coupled tables can be useful. Below you find three tables in a rather typical structure: one



transaction table and two dimension tables associating to it via one field each.

Time		Trans			Product	
Year	Date	Date	Prod	Amount	Proc	ProdGrp
1999	1999-11-30	1999-11-30	A	10	A	X
1999	1999-12-01	1999-11-30	E	14	B	X
2000	2000-01-15	1999-12-01	A	15	C	X
2000	2000-01-22	1999-12-01	B	11	D	Y
		2000-01-15	C	17	E	Z
		2000-01-15	D	13		
		2000-01-22	B	16		
		2000-01-22	C	12		

Now, let us say you want a pivot table showing sales per year and product group. If we just create one beside two list boxes showing the dimension fields it will look like this:

Sum(Amount)			Year	ProdGrp
Year	ProdGrp	Sum(Amount)	1999	X
1999	X	36	2000	Y
	Z	14		Z
		50		
2000	X	45		
	Y	13		
		58		
		108		

Even though this is a correct pivot table, the effects of QlikView's logic could here potentially lead to undesired results. If we select the year 2000 the layout will look as follows:

Sum(Amount)			Year	ProdGrp
Year	ProdGrp	Sum(Amount)	1999	X
2000	X	45	2000	Y
	Y	13		Z
		58		
		58		

Product group Z is "gone". This is natural since the value Z in the field *ProdGrp* has been excluded by the selection of the value 2000 in the field *Year*. However, the boss is likely to want Z in the chart with a 0 in the *sum(Amount)* column, so that it is clear to everyone that product group Z exists and that nothing has been sold in 2000.

You could argue that the two fields *Year* and *ProdGrp* really have nothing to do with each other and therefore should not interact just because they happen to be associated via the *Trans* table. So let's fix that by declaring the *Trans*

table as loosely coupled. Immediately our little layout will have the following look of:

Sum(Amount)			Year	ProdGrp
Year	ProdGrp	Sum(Amount)	1999	X
2000	X	45	2000	Y
	Y	13		Z
	Z	0		
		58		
		58		

The table looks like we want it to look. Note that the selection in the *Year* list box does not render any value in the *ProdGrp* list box excluded.

As a summary can be said that this situation, with one or more transaction tables surrounded by a number of dimensions which you do not want to exclude, is quite common. Loosely coupled tables is one possible way of handling such cases.

Loosely Couples Tables and Subtotals

When using loosely coupled tables in combination with non-hierarchical dimension data subtotals in pivot tables may become incorrect. The example below is very similar to the previous example, but product B now belongs to two product groups, X and Y. The product D has disappeared and that sale is now product B instead.

Time		Trans			Product	
Year	Date	Date	Prod	Amount	Prod	ProdGrp
1999	1999-11-30	1999-11-30	A	10	A	X
1999	1999-12-01	1999-11-30	E	14	B	X
2000	2000-01-15	1999-12-01	A	15	B	Y
2000	2000-01-22	1999-12-01	B	11	C	X
		2000-01-15	B	13	E	Z
		2000-01-15	C	17		
		2000-01-22	B	16		
		2000-01-22	C	12		

This means that the total amount sold is the same, which can be seen if we take a look at the pivot table, now expanded with *Prod* as a third dimension:

Sum(Amount)			
Year	ProdGrp	Prod	Sum(Amount)
1999	X	A	25
		B	11
	Y	B	11
	Z	E	14
			50
2000	X	B	29
		C	29
	Y	B	29
			58
			108

As you can see from the absence of product group Z in the year 2000 we have not set any table loosely coupled. QlikView handles all subtotals correctly, i.e. the two occurrences of B under each year are only counted once in the subtotals. This is an effect of QlikView's normal internal logic.

Now let us loosen the *Trans* table as we did before. The pivot table will look as follows:

Sum(Amount)			
Year	ProdGrp	Prod	Sum(Amount)
1999	X	A	25
		B	11
		C	0
	Y	B	11
	Z	E	14
			61
2000	X	A	0
		B	29
		C	29
	Y	B	29
	Z	E	0
			87
		148	

By cutting the association within the *Trans* table there is now no way for QlikView to keep track of what has been counted before in the subtotals. The two occurrences of B will be counted twice and the subtotals are too big.

This situation should be avoided, which leads to the conclusion that loosely coupled in combination with non-hierarchical dimensions should be used with care.

Note The **Total Mode** (as set on the chart's **Expressions** page) is set to **Expression Total** (default) in all the cases above. If you use the **Sum of Rows** mode there is no difference between the two cases.

13 BOOKMARKS

13.1 About bookmarks

The current state of selections can be saved as bookmarks for later use. There are different types of bookmarks:

Document bookmarks

are stored inside the qvw document. They will always be available to whomever opens the document locally or from a QlikView Server.

User bookmarks

are stored separately on the user's computer. They will only be available to the user who created them on the computer where they were created. If the document is moved or renamed all personal bookmarks related to it will be lost.

Personal server bookmarks

are only available when working with a document on QlikView server and only to authenticated users. They are stored in a repository on the server and are accessible to the user from any computer where authenticated.

Shared server bookmarks

are only available when working with a document on QlikView Server and only to authenticated users. Any user who has created a personal server bookmark may flag this as shared to other users. They will then become available to other users. Just like personal server bookmarks shared server bookmarks are stored in a repository on the server.

13.2 Creating bookmarks

The current state of selections can be saved as a bookmark by choosing **Add Bookmark** from the **Bookmarks** menu or by pressing CTRL+B. Bookmarks can also be created via bookmark objects in the QlikView layout. Doing this will open the **Add Bookmarks** dialog. The dialog looks slightly different depending on whether you are working with a local document or with a document on QlikView Server. We start by

describing the dialog for local documents and then describe the differences for server documents.

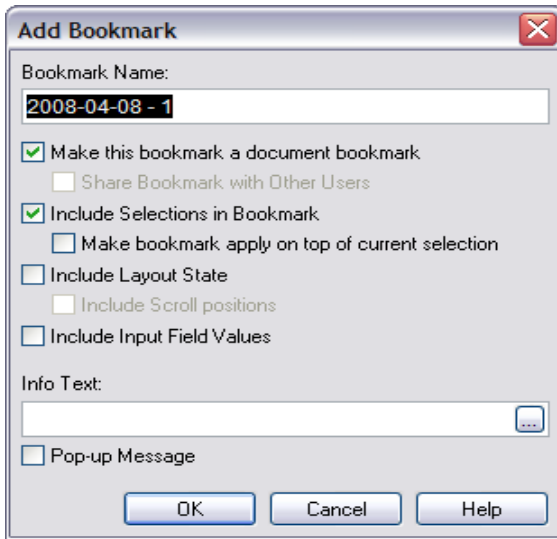


Figure 26. The Add Bookmark dialog for local documents

Bookmark Name

The default name for the created bookmark is that of the current date. In addition, the first bookmark created on a specific day gets number 1, the second number 2, etc. However, you can change the default name to a more explanatory text by entering a name of your choice.

Make this bookmark a document bookmark

By marking this check box, the bookmark will be saved with the document. This option is available for local documents only. If the check box is not marked, the bookmark will be created as a user bookmark, stored locally on your computer.

Include Selections in Bookmark

By marking this check box, the bookmark will include the selections made in the application.

Make bookmark apply on top of current selection

By marking this check box, the bookmark will be applied without first clearing existing selections in the document.

Include Layout State

By marking this check box, the bookmark will store information about the sheet from which it was created and the state of all sheet objects on that sheet at the time of the bookmark's creation, including, for example, the expanded or collapsed state of a pivot table and the cycling position of a chart. When recalling a bookmark with layout information, QlikView will attempt to activate the sheet and restore sheet objects to that state.

Include Scroll Positions

Marking this check box will include the current scroll positions in the bookmark.

Include Input Field Values

By marking this check box, the values in Input Fields will be stored in the Bookmark.

Info Text

In the edit box you may enter a text describing the bookmark or a message to be shown when recalling the bookmark.

Pop-up Message

If this check box is marked, the bookmark info text (if it exists) will be displayed in a pop-up window each time the bookmark is recalled.

The differences for server documents are as follows:

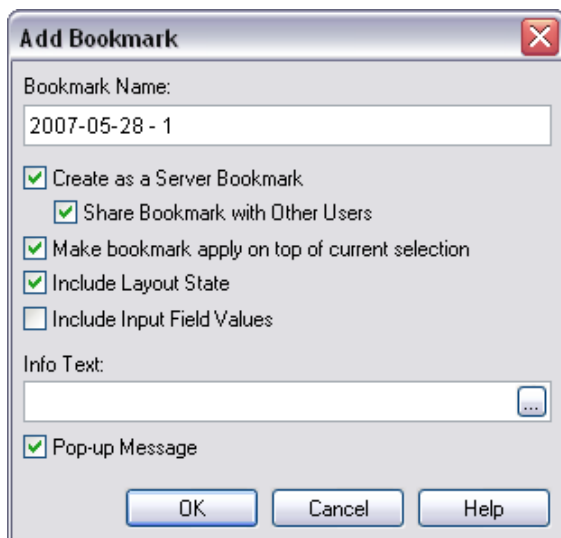


Figure 27. The Add Bookmark dialog for server documents

Create as Server Bookmark

When this check box is marked, the bookmark will be created as a personal server bookmark, stored in a repository on the server. This option is available only if you are an authenticated user and provided that the QlikView Server and the document allows you to create server bookmarks.

Share Bookmark with Other Users

Provided you can create a personal server bookmark, marking this check box will immediately share the bookmark for use by other authenticated users of the same server document. You may at any time revoke the sharing by deselecting the **Share** check box in the **My Server Bookmarks** page of the Bookmarks dialog.

13.3 Applying bookmarks

The **Bookmarks** menu contains a list of the ten first document bookmarks and the ten first personal bookmarks. To apply a bookmark, simply select it in the list.

Further bookmarks are listed in the **Bookmarks** dialog, which is reached with the **More** command in the **Bookmarks** menu.

Bookmarks can also be selected in the drop-down list of a bookmark object.

13.4 Bookmarks dialog

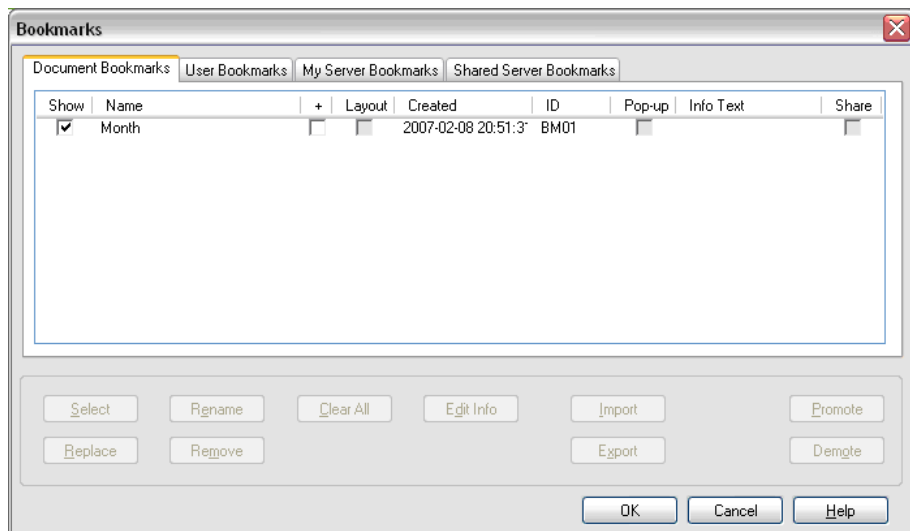


Figure 28. The Bookmarks dialog with four tabs, as it appears when working with a server document

The **Bookmarks** dialog is divided into two pages (local documents), or five pages (server documents), one for document bookmarks, one for user bookmarks, one for your personal server bookmarks (server documents only), one for other users' shared bookmarks (server documents only) and one for temporary bookmarks (server documents only).

At the top of the dialog you will find a list of all bookmarks currently defined in the QlikView document. The bookmarks can be sorted by several columns:

Show

Mark this check box if you want the bookmark to appear in bookmark object listings and in the Bookmarks menu. If the check box is left unchecked, the bookmark will not appear in those places, but will remain accessible via this dialog.

Name

Name of the bookmarks. To sort the bookmarks in alphabetical order, click the header of the column. If you click twice, the bookmarks are sorted in reverse alphabetical order.

+

If the check box is marked for a bookmark, that bookmark will be applied without first clearing existing selections in the document. A

bookmark can only be applied on top of existing selections if they do not conflict with the selections in the document.

Layout

If the check box is marked for a bookmark, the bookmark contains information about the sheet from which it was created and the state of all sheet objects on that sheet at the time of the bookmark's creation. When recalling a bookmark with layout information, QlikView will attempt to activate the sheet and restore sheet objects to that state. This check box can be toggled on/off for a bookmark containing layout state but never selected for a bookmark created without layout information. If deselected for a bookmark containing layout state, no attempt to restore layout will be made when recalling the bookmark.

Created

Information on when the bookmarks were created. To sort the bookmarks by their date of creation, click the header of the column. If you click twice, the last created bookmark is displayed at the top.

ID

The unique ID of the bookmark. Upon creation, every bookmark is assigned a unique ID for control via Automation. The first bookmark of each type in a document will be assigned the ID BM01. This means you could have a document bookmark and a server bookmark with the same ID.

A bookmark ID may be changed via Automation to any other string that is currently not used as ID for any other bookmark, sheet or sheet object in the document. Renaming the bookmark does not change the bookmark ID. Exporting and then importing a bookmark will create a new bookmark ID. The bookmark ID may be preceded by a prefix describing its type (document, user etc.).

Pop-up

If this check box is marked for a bookmark, the bookmark info text (if it exists) will be displayed in a pop-up window each time the bookmark is recalled. The pop-up window is closed by clicking anywhere in the QlikView application window. You can also set a time on the **General page** of the **User Preferences** dialog after which bookmark pop-up windows will automatically close.

Info Text

A text that can either be used as a general comment for the bookmark or be displayed in a pop-up window each time the bookmark is recalled.

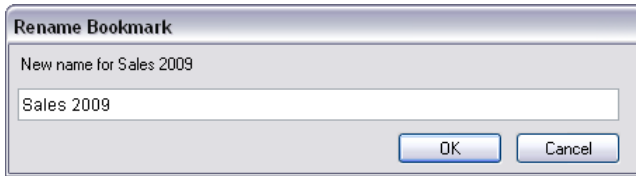
Share

This check box is only available in the **My Server Bookmarks** page. By marking it for one of your personal server bookmarks, you will make that bookmark available to other authenticated users of the same server document in their **Shared Server Bookmarks** page. The bookmark will stay in your **My Server Bookmarks** page and will not appear in your **Shared Server Bookmarks** page. You may at any time revoke the sharing by deselecting the check box.

At the bottom of the dialog you will find a number of buttons which can be used to perform actions with the bookmark currently selected in the list above.

Select

Displays the selected bookmark on the screen.

Rename

Opens the **Rename Bookmark** dialog where you may specify a new name for the bookmark. On pressing **OK** you will come back to the **Bookmarks** dialog. You can only rename the bookmarks that you have created yourself.

Replace

Replaces the selected bookmark with the current selections. You can only rename the bookmarks that you have created yourself.

Remove

Removes the selected bookmark. You can only remove the bookmarks that you have created yourself.

Clear All

Removes all bookmarks in the current pane of the dialog. You can only remove the bookmarks that you have created yourself.

Set Clear State

Click this button if you want to use the selection of the highlighted bookmark as clear state.

Edit Info

Opens a dialog for editing the bookmark's info text. This text can either be used as a general comment for the bookmark or be applied

in a pop-up window each time the bookmark is recalled. You can only edit the bookmarks that you have created yourself.

Import

Allows you to select a QlikView bookmark file and then opens the **Import Bookmark(s)** dialog (see below).

Export

Opens the **Export Bookmark(s)** dialog (see below) where you can export selected bookmarks to a QlikView bookmark file.

Promote

Moves the selected bookmark one step up in the list. This can also be achieved by pointing at a bookmark with the mouse and dragging it to any position in the list.

Demote

Moves the selected bookmark one step down in the list. This can also be achieved by pointing at a bookmark with the mouse and dragging it to any position in the list.

Move Local User Bookmarks to Server

This text only appears when you work with a server document and in the **User Bookmarks** tab. By clicking on the text you can convert all your local user bookmarks to server bookmarks, stored in the server repository. This is a recommended practice as your personal server bookmarks will stay accessible via the server even if you change computer or if the server document is renamed. You will be asked to confirm before the conversion takes place. The conversion is all-or-nothing (within the active document) and one-way only.

OK

Accepts the changes and closes the dialog.

Cancel

Closes the dialog without saving the changes.

Help

Displays help on bookmarks.

13.5 Exporting Bookmark(s)

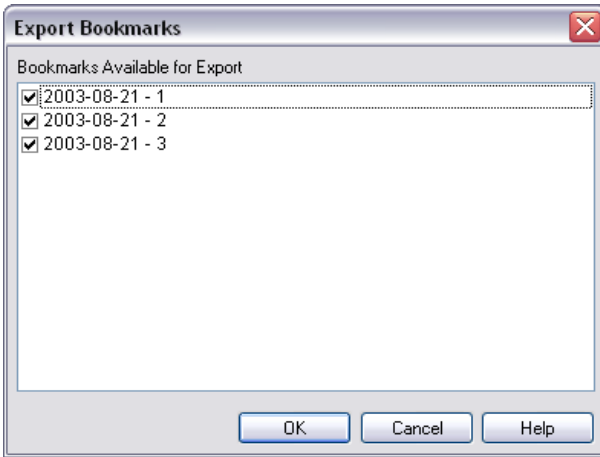


Figure 29. The Export Bookmark(s) dialog

Bookmarks are exported to a bookmark file by choosing **Export** from the **Bookmarks** menu, which opens the **Export Bookmark(s)** dialog (see figure above). In the **Export Bookmark** dialog you see all bookmarks which were defined in the current document. To the left of each bookmark name in the list is a check box, which is by default selected. By deselecting the check box for one or more bookmarks, you exclude these bookmarks from the export.

On clicking **OK** you will be asked to choose a file name and location for the bookmark file. Once saved the bookmark file can be kept for later use or sent to other QlikView users using the same application.

13.6 Importing Bookmark(s)

Bookmarks are imported to a bookmark file by choosing **Import...** from the **Bookmarks** menu, which opens the **Import Bookmark(s)** dialog (see figure below).

You will first be taken to a file browse dialog to open a QlikView bookmark file. After selecting a bookmark file (.qbm extension) you will be taken to the **Import Bookmark** dialog.

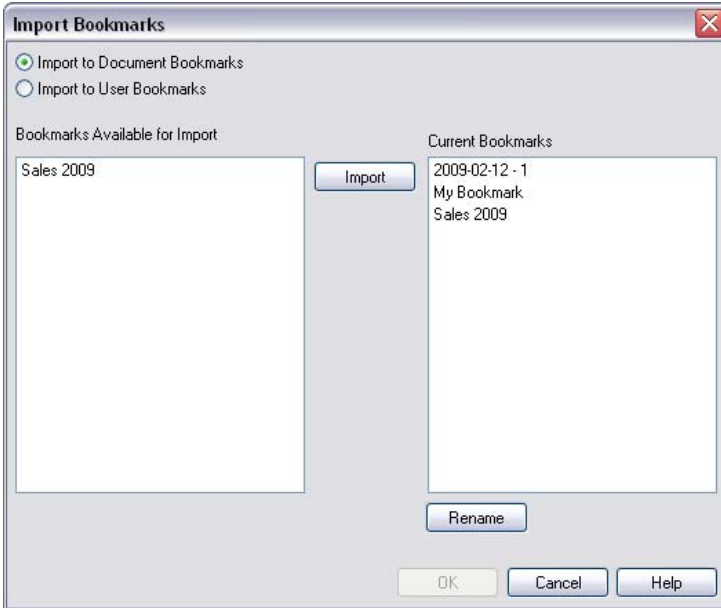


Figure 30. The *Import Bookmark(s)* dialog

Bookmarks Available for Import

To the left you see all bookmarks which were defined in the bookmark file.

Current Bookmarks

To the right you see a list of existing user bookmarks or document bookmarks in the document. Bookmarks must be imported into either the user bookmarks domain or into the document bookmarks domain. To choose between showing user bookmarks and document bookmarks use the following two controls:

Import to Document Bookmarks

Mark this radio button to import bookmarks into the document bookmark domain and to show document bookmarks in the **Current Bookmarks list**. This option is only available when working in local documents.

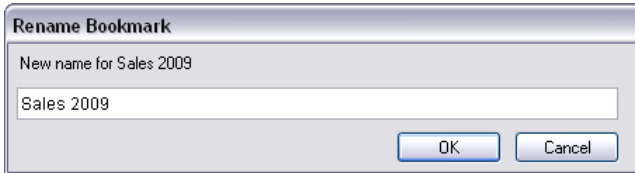
Import to User Bookmarks

Mark this radio button to import bookmarks into the user bookmark domain and to show user bookmarks in the **Current Bookmarks** list.

Import

Mark one or more bookmarks in the **Bookmarks Available for Import** list and press this button to import them into the **Current Bookmarks** list.

Rename



Before importing a bookmark with the same name as an existing bookmark you must first rename the existing bookmark. Select the existing bookmark in the right list by clicking on it and then press the **Rename** button. In the dialog **Rename Bookmark** that appears you may then specify a new name. On pressing **OK** you will come back to the **Import Bookmark(s)** dialog with the bookmark appearing renamed in the list.

Imported bookmarks will of course only be meaningful if the document to which they are imported contains the fields and field values to which the bookmarks refer.



14 REPORTS

14.1 About Reports

Printing a report often means printing a single table or graph. In QlikView this is done as easily as selecting a sheet object and then choosing **Print** from a menu or toolbar. Sometimes however, there is a need for producing more complex reports including multiple charts and/or tables. This is where the QlikView **Report Editor** comes into the picture. The QlikView **Report Editor** offers the possibility to group a number of different sheet objects together on one or more pages with full control of page layout, including headers/footers etc.

QlikView reports come in two flavors, document reports and user reports.

Document Reports

Document reports are created with the QlikView document and stored as part of the QVW file. Any user accessing the QlikView document locally or via QlikView Server can access the document reports in the document.

User Reports

User reports are created by a user working with a document from QlikView Server. The report is stored on the local client machine in a manner similar to that of user bookmarks. Only the local user has access to his/her own user reports. User reports can only be created from the Windows based QlikView clients (not from Java or Zero-footprint clients).

My Server Reports

My Server Reports are only available when working with a document on QlikView Server and only to authenticated users. They are stored in a repository on the server and are accessible to the user from any computer where authenticated.

Shared Server Reports

Shared server reports are only available when working with a document on QlikView Server and only to authenticated users. Any user who has created a personal server report may flag this as shared to other users. It will then become available to other users. Just like My Server Reports, shared server reports are stored in a repository on the server.

Note When using the term “Report Generator“ one usually refers to a well defined category of software products. These typically work by combining data from several SQL queries (more or less obscured

from the user by means of graphical interfaces), each of which is formatted for printing in different ways. QlikView reports fetch their data from QlikView sheet objects and lack the possibility to draw data from SQL queries. Just as it is absolutely correct to say that QlikView can produce reports, it is incorrect to label the QlikView Report Editor a report generator in the traditional sense.

14.2 The Reports Menu

The **Reports** menu lists all available existing reports for immediate selection. At the top all document reports are listed followed by user reports for the current document. When clicking one of the report names the **Print** dialog will be opened for printing the report. The reports will be greyed out if no printer is installed on the local computer or on the server if the document is opened in server.

Select **Edit Reports...** or click on the icon in the toolbar to create new reports or edit existing reports in the **Report Editor** dialog.



14.3 The Report Editor Dialog

The **Report Editor** dialog consists of two pages, the **Report List** and the **Page Editor**. When first entering the report editor, the **Report List** will be shown.

Report List

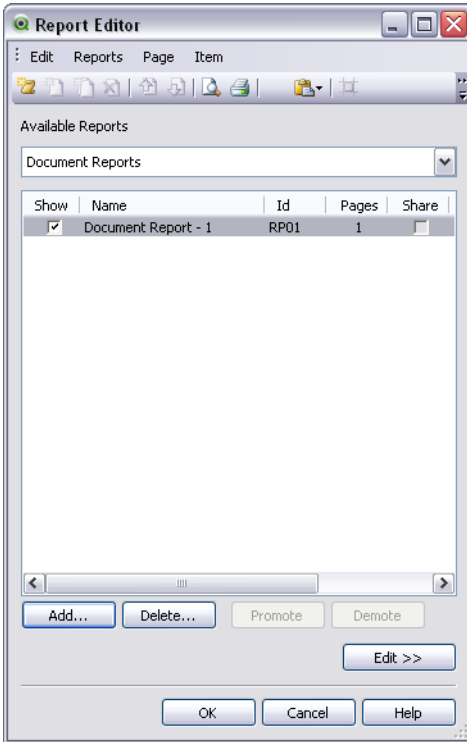


Figure 31. The Report List page of the Report Editor dialog

The **Report List** page is used to manage reports. You may create new reports and remove existing reports. This is also where you select a report for page editing in the **Page Editor**.

List of Reports

At the top of the page you will find a drop-down and a list of reports. The drop-down is used for selecting which type of reports should be shown in the list. For local documents you may choose between **Document Reports** or **User Reports**. For server documents you may choose between **My Server Reports**, **Shared Reports** or **User Reports**. The list itself includes several columns:

Show

Mark this check box if you want the report to appear in the Reports menu. If the check box is left unchecked, the report

will not appear but will still remain accessible via this dialog.

Name

Name of the report.

ID

The unique report ID (see below).

Pages

The number of pages currently defined in the report.

Share

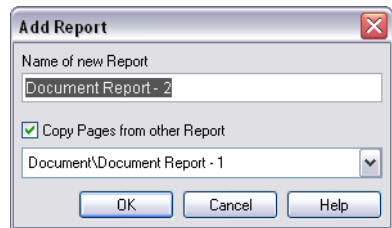
This check box is only available in the My Server Reports list. By marking it for one of the My Server Reports, you will make that report available to other authenticated users of the same server document in their Shared Server Reports list. The report will stay in your My Server Reports list and will not appear in your Shared Server Reports list. You may at any time revoke the sharing by deselecting the check box.

Author

The authenticated name of the creator of the report.

Add...

Push this button to create a new report. If **Document Reports** is selected above the **List of Reports**, the new report will be a document report. If **User Reports** is selected, the new report will be a user report. The dialog



New Report is opened so that you can name your report. Mark the **Copy Pages from Other Report** and pick an existing report in the drop-down menu to copy pages from that report.

Delete

Push this button to remove the report currently selected in the **List of Reports**.

Promote

Push this button to promote the report currently selected in the **List of Reports** one step up in the list.

Demote

Push this button to demote the report currently selected in the **List of Reports** one step down in the list.

Move Local User Reports to Server

This text only appears when you work with a server document and if you have selected **User Reports** in the drop-down at the top of the list. By clicking on the text you can convert all your local user reports to server reports and store them in the server repository. This is a recommended practice as your My Server Reports will stay accessible via the server even if you change computers or if the server document is renamed. You will be asked to confirm before the conversion takes place. The conversion is all-or-nothing (within the active document) and one-way only.

Edit >>

Invokes the **Page Editor** for the selected report. Pushing this button is equivalent to double-clicking a report in the **List of Reports**.

OK

Saves changes and closes the dialog.

Cancel

Closes the dialog without saving changes.

Help

Opens online help.

Pages

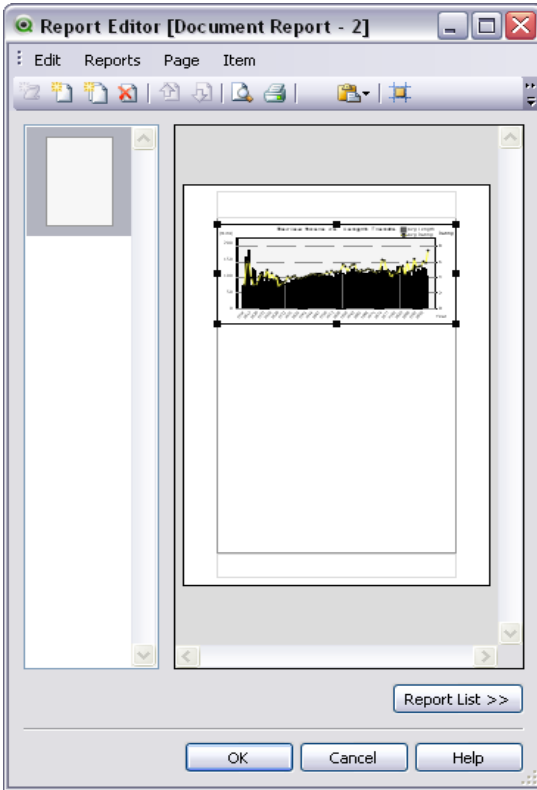


Figure 32. The Page Editor page of the Report Editor dialog

The **Page Editor** page is used to define the pages of a report selected on the **Report List** page. The page has two panes and a toolbar to assist you in designing report pages. At the bottom you find the following buttons:

Report List>>

Takes you back to the **List of Reports** page.

OK

Saves changes and closes the **Report Editor** dialog.

Cancel

Closes the **Report Editor** dialog without saving changes.

Help

Invokes on-line help.

Edit**Copy**

Copies the currently selected item and the current page in the report. Also available as right-click command in the page preview pane.

Paste Report

Pastes the report from the Clipboard.

Paste Page

Pastes the page from the Clipboard.

Paste Item

Pastes the item from the Clipboard. Also available as right-click command in the page preview pane.

Snap to Grid

Displays a grid in the report to easier align the report items.

Reports**Add**

Adds a report. Only available when viewing the **Report List**.

Delete

Deletes a report. Only available when viewing the **Report List**.

Promote

Moves the selected report up one step in the list of reports. Only available when viewing the **Report List**.

Demote

Moves the selected report down one step in the list of reports. Only available when viewing the **Report List**.

Export

Saves the report file as an XML document. Only available when viewing the **Report List**.

Import

Imports a report that has been saved as XML. Only available when viewing the **Report List**.

Print Preview...

Opens the **Print Preview** dialog and where you can see how the current report will look when printed with the current selections in the QlikView document.

Print...

Opens the **Print** dialog and prints the current report.

Report Settings...

Opens the **Report Settings** dialog (see page 173 for details) where you can set various properties for the report. This includes margins, header/footer setting etc.

Page**Promote**

Promotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.

Demote

Demotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane

Add Multi Page

Adds a multi-paper page after the currently selected page.

Add Single Page

Adds a single-paper page after the currently selected page.

Delete

Deletes the currently selected page.

Page Settings...

Opens the **Page Settings** dialog (see page 178 for details) where you can set various properties for the selected page. Also available as right-click command when no print object is selected and you right-click the page background.

Item**Align Top**

Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the top. Also available as right-click command in the page preview pane.

Align Bottom

Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the bottom. Also available as right-click command in the page preview pane.

Align Left

Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the left. Also available as right-click command in the page preview pane.

Align Right

Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the right. Also available as right-click command in the page preview pane.

Same Width

Gives the currently selected items the same width. Also available as right-click command in the page preview pane.

Same Height

Gives the currently selected items the same height. Also available as right-click command in the page preview pane.

New Image...

Creates a new text object (in a hidden location in the document) and then opens the **Text Object Properties** dialog for selection of an image. The image will appear as a regular print item in the page preview pane, where it can be moved and sized just like any other print item.

New Text...

Creates a new text object (in a hidden location in the document) and then opens the **Text Object Properties** dialog where the properties for the new text object can be edited. The text object will appear as a regular print item in the page preview pane, where it can be moved and sized just like any other print item.

New Current Selections...

Creates a selection stamp object for the report. The selection stamp will appear as a regular print item in the page preview pane, where it can be moved and sized just like any other print item.

Delete

Deletes the currently selected item(s). Also available as right-click command in the page preview pane.

Item Settings...

Opens the **Item Settings** dialog (see page 180 for details) where you can set various properties for the selected print

item. See page 180 for a detailed documentation of this dialog. Also available as right-click command in the page preview pane.

Page List Panes

To the left you will find a list of all pages in the report. QlikView reports can contain two different types of pages, single-paper pages and multi-paper pages. Any number of pages can be added and the two types may be mixed in any combination within a report.

Single-Paper Pages

A single paper can contain any number of sheet objects. The page will always be printed on exactly one paper page (or PDF page) and the sheet objects will have to be zoomed or truncated to fit the page. Sheet objects may overlap on the page. Extra text can be added.

Multi-Paper Pages

A multi-paper page can contain one sheet object which may spread over any number of paper pages (or PDF pages), depending on the amount of data to be printed. A multi-paper page is the typical choice when printing large tables. In addition to the dynamic size objects you may add fixed size objects in introduction and/or appendix zones on the page. These zones work just like single-paper pages.

Manipulating the page list

You select a page for editing by clicking on it in the list. There are a number of toolbar commands directly associated with the page list pane:

Add Single-Paper Page

Adds a single-paper page after the currently selected page.



Add Multi-Paper Page

Adds a multi-paper page after the currently selected page.



Delete Page

Deletes the currently selected page.



Promote Page

Promotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.



Demote Page

Demotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.

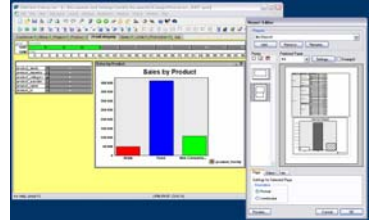


Page Preview Pane

To the right of the page list pane you will find a preview of the page currently selected in the page list pane.

Adding Sheet Objects to the Report

Simply drag any sheet object from the layout into the report page preview. The print objects will appear in the preview as they currently look in the QlikView layout. The exact appearance of the object in the report will of course dynamically reflect how the sheet object looks in the layout at the time of printing, with respect to property changes as well as to changes in selected data. It is possible to select multiple sheet objects in the layout and drag them as a group into the report. Note that only one print object can be added to the central dynamic area of a multi-paper page!

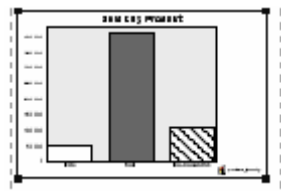


Navigating and Selecting Print Objects on Page

You select a print object by clicking it. Once you have selected one print object, you may move the selection to the next object using the TAB key (SHIFT+TAB to move backwards). You may use CTRL-click to select multiple objects. In multi-paper pages you may drag print objects between the different areas.

Sizing and Positioning Print Objects on Page

Once added the sheet object will appear in the page preview pane. There will be a border around the object and placeholders in each corner to mark that the object is selected. Point at it with the mouse and move it to the desired position. Use the corner placeholders to adjust the size. A dashed line shows the current margins. No print object can be placed outside these margins.



Use the corner placeholders to adjust the size. A dashed line shows the current margins. No print object can be placed outside these margins.



Toolbar Commands

There are two toolbar commands directly associated with the page preview pane:

Design Grid

Toggles design grid on and off. This also activates the snap-to-grid functionality. With design grid and snap-to-grid turned on it is easier to align the sheet objects on the page, but you have less exact control.



Zoom

In this drop-down you may specify a zoom factor for the entire preview pane. With a larger zoom factor it becomes much easier to do exact placement of objects. You may have to use scroll bars to navigate to the different parts of the page in the preview pane.

Then, there are some toolbar buttons for printing the report from inside the **Report Editor** dialog.

Print Preview...

Push this button to open the **Print Preview** dialog and see how the current report will look when printed with the current selections in the QlikView document. This button is also available from the **Page Editor** page of the **Report Editor** dialog.



Print...

Push this button to open the **Print** dialog and print the current report. This button is also available from the **Page Editor** page of the **Report Editor** dialog.



Finally, there is a toolbar button for pasting copied reports, pages and items.



Report Settings Dialog

In this dialog you will find global (non-page related) settings for the report. The dialog has four pages.

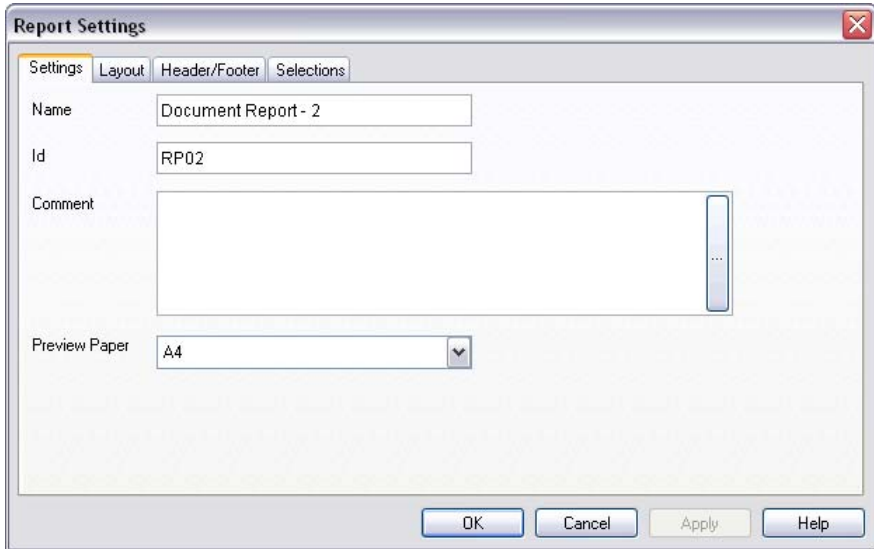


Figure 33. The Settings page of the Report Settings dialog

Settings

This page contains basic properties for the report:

Name

Name of the report. This may be any text string. The name may be given as a calculated formula for dynamic update.

Report ID

The unique ID of the report. Upon creation, every QlikView layout entity, including reports, is assigned a unique ID for control via Automation. The ID by default consists of a combination of letters defining the type of entity, as well as of a number. The first report of a document will be assigned the ID RP01. An ID may be changed to any other string that is currently not used for any other report, sheet object, sheet, bookmark or alert in the document.

Comment

This is a commentary field where the creator of a report can describe the purpose of the report or any details with regard to the report. It is not used outside of this dialog.

Preferred Paper Size

In the drop-down you can specify the size of paper intended for the report. The proportions of the

A screenshot of a drop-down menu with a light blue border and a small downward-pointing arrow on the right side. The text 'A4' is visible inside the menu.

selected paper size will be reflected in the page preview pane. If the report is ultimately printed on another paper size than the one the report was designed for, QlikView will attempt to adjust the report contents by means of zooming the print objects to fit the new paper size.

Conditional Show

If this check box is marked, the report may be dynamically shown or hidden depending on the value of a condition expression entered in the edit box to the right. The condition expression will be evaluated each time the list of available reports is generated. The report will only be available when the condition returns true. Users with ADMIN privileges for the document can override all show conditions with the **Show All Sheets and Sheet Objects** in the **Security** page of **Document Properties** (see page 42 in Book

II). This functionality can be toggled by pressing CTRL+SHIFT+S.

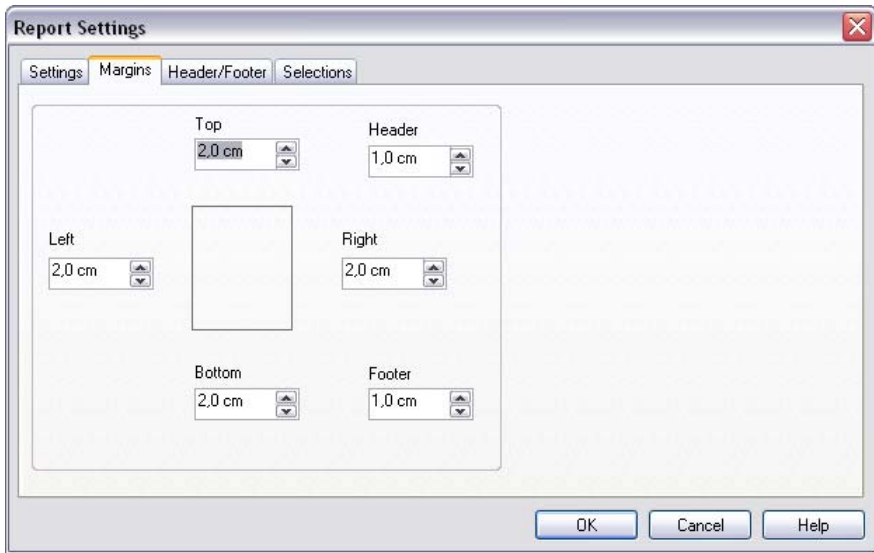


Figure 34. The Margins page of the Report Settings dialog

Margins

This page contains margin settings for the report. The controls are the same as for the **Layout** page of the **Print** dialog see page 264 in Book II.

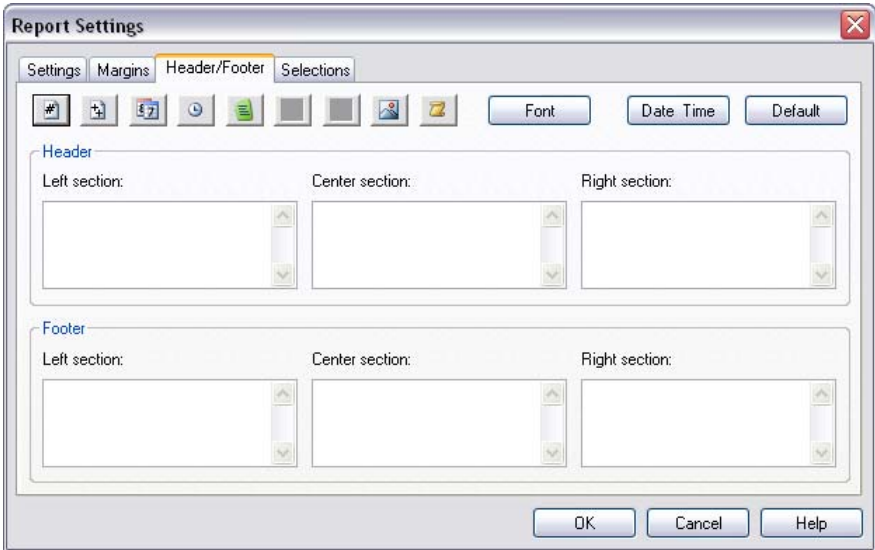


Figure 35. The Header/Footer page of the Report Settings dialog

Header/Footer

This page contains header/footer settings for the report. The controls are the same as for the **Header/Footer** page of the **Print** dialog see page 266 in Book II.

Selections

This page contains settings for the selections to be used when printing the report.

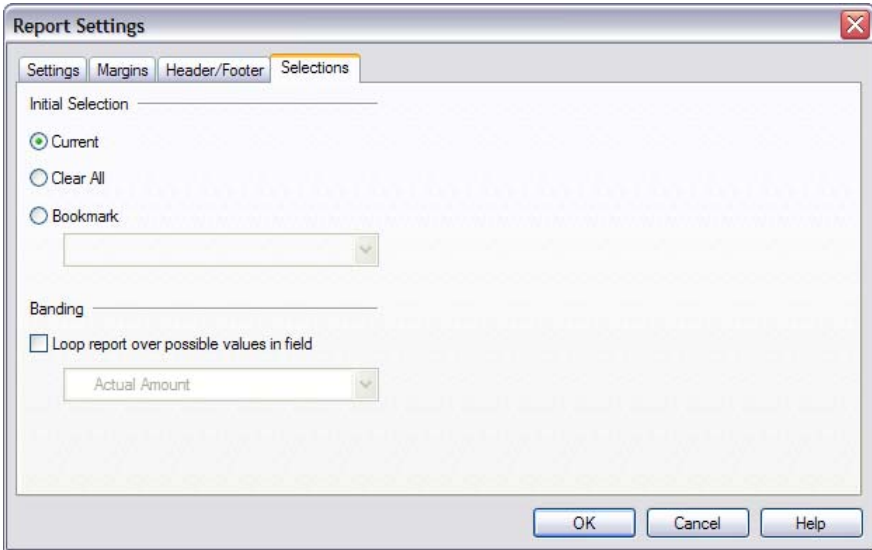


Figure 36. The Selections dialog of the Report Settings dialog

Initial Selections

These radio buttons allow you to state the initial selections for the report print. Regardless of which setting you use, selections prevailing before the report print will be re-established after the print.

Current Selections

Use current selections as the basis of the report print (default).

Clear All

All current selections in the document will be cleared before printing the report. After completed printing, the original selections will be reapplied.

Bookmark

A bookmark, which can be selected in the drop-down box will be applied before printing the report. After completed printing, the original selections will be reapplied.

Loop Report Over Possible Values in Field

When this check box is marked, the entire report will be printed repeatedly while selecting each possible value in the field specified in the drop-down box below. If there are no possible values in the specified field, nothing will be printed. After completed printing, the original selections will be reapplied. If page numbers are used, these will be continuous over all pages printed.

Page Settings Dialog

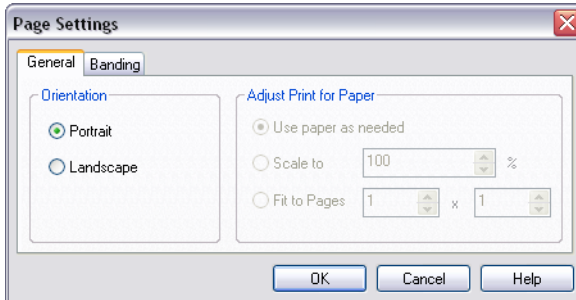


Figure 37. The General page of the Page Settings dialog

In this dialog you will find settings for the currently selected page. The dialog has two pages:

General

This page contains basic properties for the page.

Orientation

Each page can have its separate setting for orientation. Choose between **Portrait** and **Landscape**.

Adjust Print for Paper

This setting is only available for multi-paper pages.

Use Paper as Needed

No scaling of the sheet object printout. As many pages as needed in each direction will be used.

Scale to x%

The sheet object printout will be scaled to a fixed percentage of its original size. As many pages as needed in each direction will be used.

Fit to X by Y

The sheet object printout will be scaled to fit on a fixed number of pages.

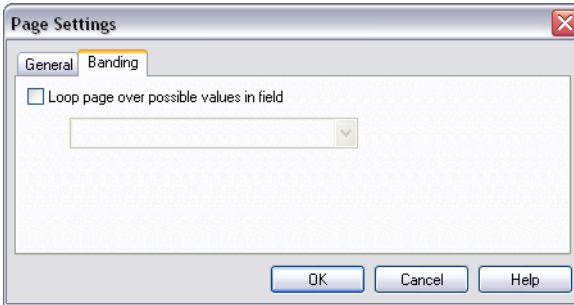


Figure 38. The Banding page of the Page Settings dialog

Banding

This page contains properties for banded output of the page. This means that the page is printed repeatedly for each possible value in a specified field.

Loop Page Over Possible Values in Field

When this check box is marked, the page will be printed repeatedly while selecting each possible value in the field specified in the drop-down box below. If there are no possible values in the specified field, nothing will be printed. Note that if a selection loop has been specified also on the report level (see the **Selections** page of the **Report Settings** dialog described above) you may face a situation where the report selection excludes all values in the page level loop variable. The page will then be skipped for that report loop value. After completed printing, the original selections will be reapplied. If page numbers are used, these will be continuous over all pages printed.

Item Settings Dialog

In this dialog you will find settings for the print item (object) currently selected in the page preview pane..

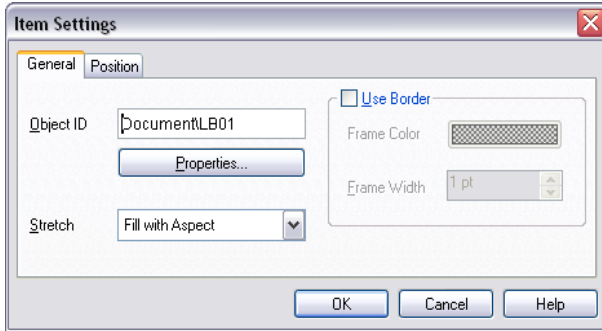


Figure 39. The General page of the Item Settings dialog

General

The General page contains settings for the print item currently selected in the page preview pane

Object ID

The sheet object ID of the sheet object to be printed.

Properties

Opens the Properties dialog for the sheet object associated with the print item. Changes made in this dialog will be applied on the actual sheet object and of course reflected in the report print result.

Stretch

This control is only available for objects on single-paper pages. Here you may select how the print object should be sized to fit inside the rectangle assigned to it.

Clip

Do not size the print object. If it is too large for the placeholder rectangle it will be truncated. If it is too small there will be blank space around it.

Fill

Size the print object so that it fills the placeholder rectangle.

Fill with Aspect

Size the print object so that it fits in the placeholder rectangle while keeping the original sheet object ratio.

Use Border

The print object is normally printed on the page without any frame or border. By selecting this check box, you may have a border around the object.

Frame Color

Press this button to select the border color.

Frame Width

Here you may specify the thickness of the border.

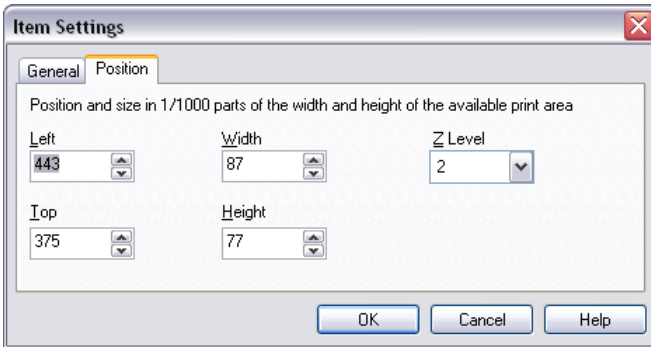


Figure 40. The Position page of the Item Settings dialog

Position

The Position page contains settings for positioning and sizing the objects in the available print area.

Left

Sets the position of the currently selected item(s) relative the left margin. 0 is to the far left of the available print area.

Top

Sets the position of the currently selected item(s) relative the top. 0 is at the top of the available print area.

Width

Sets the size of the currently selected item(s) in 1/1000 parts of the width of the availableprint area.

Height

Sets the size of the currently selected item(s) in 1/1000 parts of the height of the availableprint area.

Z Level

Sets the layer of the currently selected item(s). If items overlap the item at the lowest level (1) will be printed first, then the next layer will be printed and so on.



15 ALERTS

15.1 About alerts

An alert is a composite entity typically consisting of three basic parts:

- 1 A **condition**, i.e. a QlikView expression forming a logical condition, that can be either true or false.
- 2 A **logical** state (bookmark, clear all or current selection state) that should be applied before checking the state of the condition expression.
- 3 One or more **actions** to be performed when the condition is checked and evaluates to true. Typical actions include showing a message in a pop-up window or sending a message as an e-mail to one or more recipients. Further actions can be programmed via macros.

When an alert is checked, the condition met and the actions performed, it can be said that the alert has “fired”.

15.2 Creating alerts

The easiest way to define an alert is by using the **Alert Wizard**, which can be invoked from the **Tools** menu. Alerts may also be created and maintained in the **Alerts** dialog, which can be opened from the **Tools** menu. Alerts are stored as part of the QlikView document. Alerts can only be created and triggered from Windows versions of QlikView (not QlikView Java clients and Zero Footprint clients).

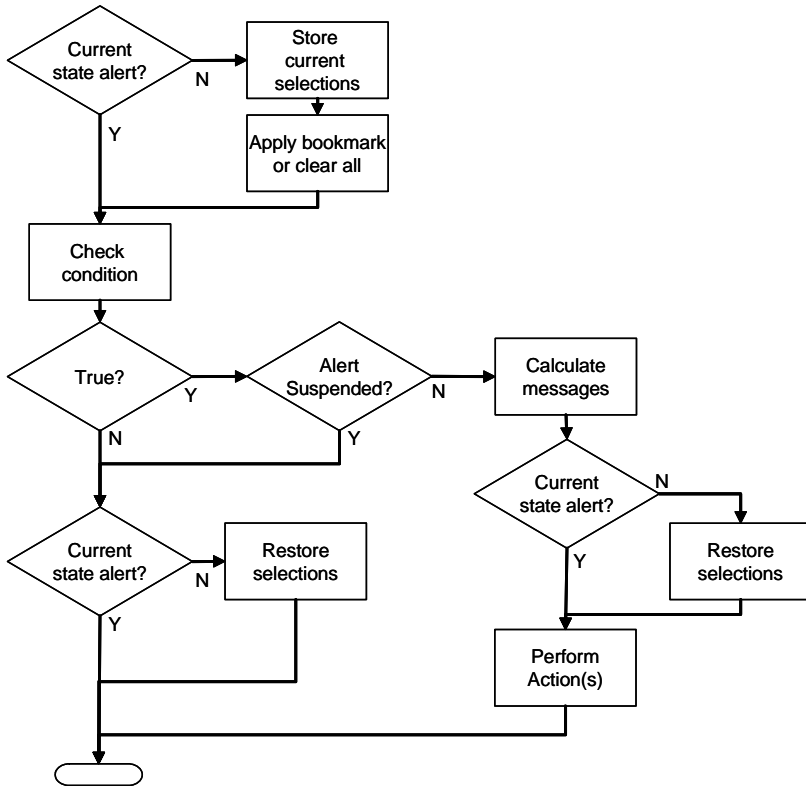
15.3 Using alerts

QlikView alert checks can be triggered in three different ways:

- 1 Automatically in the QlikView layout when there is a probability that the document’s data has changed, i.e. when the document is opened, when the script has been executed or when a Reduce Data operation has been performed.
- 2 Manually from a macro via special Automation APIs. Refer to the QlikView API guide for details.

3 External programs running QlikView in batch mode have a special Automation API to retrieve a list of fired alerts from a given context.

The chain of events following the triggering of an alert check can be shown as follows:



Alerts can be suspended after being fired until a change occurs. As an example it may be quite annoying to be reminded every time you open a document that your budget goal has been fulfilled. In that case you may set the alert to fire only once and then suspend itself until next month.

As can be seen in the flowchart, each triggering of an alert initiates a chain of logical operations in the QlikView document. Typically QlikView has to perform the following major steps:

- 1 Applying the selection relevant to the alert.
- 2 Calculating the condition expression value.

3 Restoring original selections.

The time needed to perform action 1 and 3 is of course each equal to the normal “click times“ if the selections had been applied interactively. The time needed for step 2 is the same as if the expression had resided e.g. in a sheet object. Added together, the time needed to check an alert can be quite substantial in large documents. This may be of less importance for single alerts triggered on opening or reload, but caution should be exercised when using large amounts of macro-triggered alerts, so that the document does not become sluggish.

15.4 The Alerts dialog

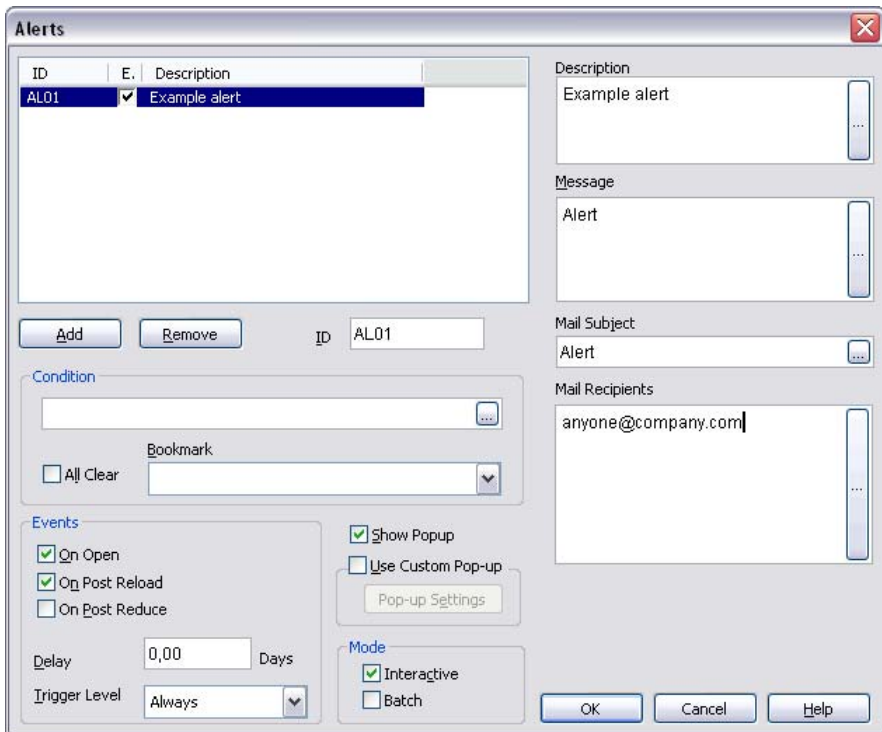


Figure 41. The Alerts dialog

List of Alerts

The top left part of the dialog contains a list with all alerts currently defined in the document. The list includes three columns:

ID

The unique alert ID (see below).

Enable

Select or deselect the check box in order to enable/disable the alert.

Comment

Description of alert as defined under **Description** (see below).

Add

Click this button to add a new alert.

Remove

Select an alert in the list and click on this button to delete it.

ID

The unique ID of the alert. Upon creation, every QlikView layout entity, including alerts, is assigned a unique ID for control via Automation. The ID by default consists of a combination of letters defining the type of entity, as well as of a number. The first alert of a document will be assigned the ID AL01. An ID may be changed to any other string that is currently not used for any other alert, sheet object, sheet, bookmark or report in the document.

Description

This is a commentary field where the creator of an alert can describe the purpose of the alert or any details with regard to the alert. It is only used in the list of alerts in this dialog.

Message

Here you type the message to be displayed with the alert. In case of pop-up alerts this is what appears in the pop-up. In case of mail alerts this is the body text of the mail. The text may be defined as a calculated expression for dynamic update (see page 249 in Book III).

Condition

A QlikView expression that should evaluate to true (non zero) when the alert is fire.

All Clear

If this check box is marked the all clear state will be used when evaluating the alert condition.

Bookmark

If the **All Clear** check box is not marked you may specify a bookmark to be applied before the alert condition is tested.

The bookmark should be stated as a bookmark ID. The bookmark must exist for the alert to work properly. If no bookmark is specified and the **All Clear** check box is left unchecked, the current selections will be used when the alert condition is tested.

Show Pop-up

Mark this check box to show the alert message as a pop-up balloon when the alert is fired.

Use Custom Pop-up

Alert pop-ups are usually formatted according to the default set under **Alert Pop-up Settings** in the **General** page of the **Document Properties** dialog. By marking this check box you may define a custom pop-up format for the selected alert.

Pop-up Settings

Click this button to open the **Pop-up Window Settings** dialog to define a custom pop-up format for the selected alert. See page 21 in Book II for details on this dialog.

Events

The alert can be manually checked by means of Automation calls at any time. Automatic checks at relevant events can be made with the settings in this group.

On Open

Mark this check box if the alert is to be automatically checked when the document is opened.

On Post Reload

Mark this check box if the alert is to be automatically checked after a **Reload** of the document (script execution). The alert will be checked also upon opening of the document if the last reload time stored in the document is later than the time noted when the alert was last checked.

On Post Reduce

Mark this check box if the alert is to be automatically checked after a data reduction (**Reduce data** command) has taken place in the document. The alert will be checked also upon opening of the document if the last reduction time stored in the document is later than the time noted when the alert was last checked.

Delay

The re-firing of automatically checked alerts may be suspended for a given time after that an alert has been fired.

Enter a number of days in this edit box to do so. Decimal numbers may be used to specify parts of a day. The value 0 indicates of course that no suspension time will be applied.

Trigger Level

The re-firing of automatically checked alerts may also be suspended on the basis of whether the alert status has changed after the firing. In this drop-down you can choose between three levels of reoccurrence.

Always

The alert is fired every time the trigger event(s) occur and the alert condition is fulfilled.

Message Changes

The firing of the alert is suspended until the alert message changes. This is of course only meaningful with dynamic message text. Note that the state of the alert condition does not necessarily have to change in order for the evaluated message to change. Any change in the **Mail Subject** will be regarded as change of message.

State Changes

The firing of the alert is suspended until the alert state changes, i.e. the alert condition has been unfulfilled during at least one check and becomes fulfilled again during a later check. This is the strongest type of alert suspension.

Mode

The alert can be defined as an interactive (automatic triggers in layout) and/or as an alert relevant for external programs running QlikView in batch mode (e.g. QlikView Publisher). If none of the check boxes in this group is marked, the alert can still be manually checked via macros.

Interactive

Mark this check box if the alert is designed for automatic triggering upon the events marked under **Events**.

Batch

Mark this check box if the alert is designed for triggering from external programs (e.g. QlikView Publisher) via the special Automation API for Batch triggers.

Mail Subject

Type the subject of the Alert mail in this field. The text may be defined as a calculated expression for dynamic update (see page 249 in Book III).

Mail Recipients

Semi-colon separated list of mail addresses. Each of these addresses will receive an e-mail message whenever the alert fires. The list may be defined as a calculated string expression for dynamic update (see “Edit Expression” on page 249 in Book III). By clicking on the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

15.5 The Alert Wizard

The Alert wizard helps you with the task of defining an alert. The process has the following basic steps:

- 1 Name/describe the alert.
- 2 Define the alert condition.
- 3 Define when to test the alert condition.
- 4 Define a possible delay before checking the alert again after firing.
- 5 Define the action(s) to take when the alert fires

Depending on selections some of the above steps may be skipped.

Starting the Alert Wizard

You start the alert wizard by choosing **Alert Wizard** from the **Tools** menu.



Figure 42. The first start page of the Alert wizard

The first time you start the wizard you will be met by two start pages, the first one describing the concept of alerts and the second one outlining the wizard and the basic steps involved. If you want to skip the start pages when you use the wizard in the future, mark the **Don't show this page again** check box in either or both pages.

Click **Next** to continue.



Figure 43. The second start page of the Alert wizard

Step 1 - Name/Describe the Alert

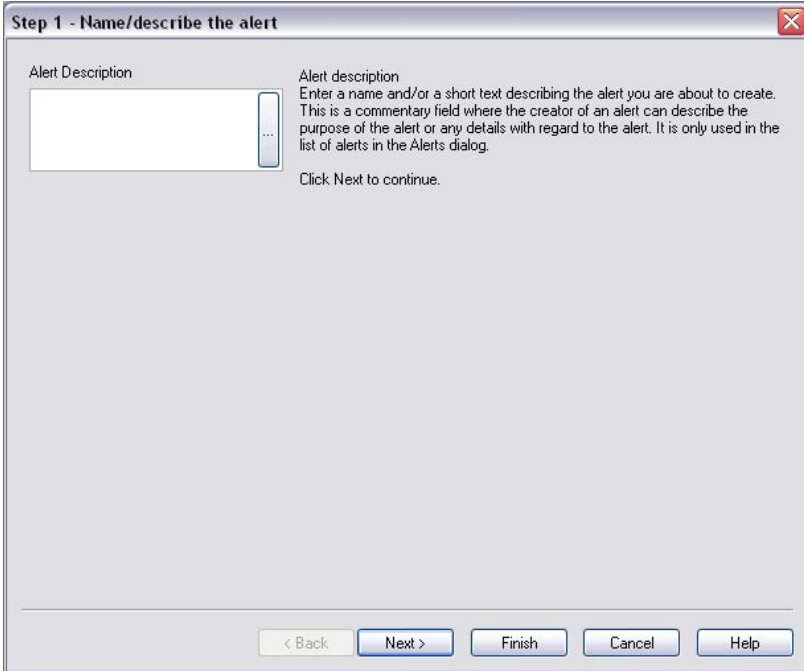


Figure 44. The first page of the Alert wizard

Enter a name and/or a short text under **Alert Description**, describing the alert you are about to create. This is a commentary field where the creator of an alert can describe the purpose of the alert or any details with regard to the alert. It is only used in the list of alerts in the Alerts dialog.

Click **Next** to continue.

Step 2 - Define the Alert Condition

Step 2 - Define the alert condition

Alert Condition

Current Selection
 All Clear
 Bookmark

In this step you define the alert condition and the selection state that should be used when checking the alert condition.

Alert condition
Enter a QlikView expression that should evaluate to true (non zero) when the alert is to fire.

Selection state when testing alert
Here you decide which selection state should be used when testing the result of the alert condition. There are three choices:

Current Selection
The alert condition will be tested using whatever selections are made at the time of checking the alert.

All Clear
All selections will be temporarily cleared before testing the alert condition.

Bookmark
A specified bookmark will temporarily be applied before testing the alert condition.

Click Next to continue.

< Back Next > Finish Cancel Help

Figure 45. The second page of the Alert wizard

In this step you define the alert condition and the selection state that should be used when checking the alert condition.

Alert Condition

Enter a QlikView expression that should evaluate to true (non zero) when the alert is to fire.

Selection State when Testing Alert

Here you decide which selection state should be used when testing the result of the alert condition. There are three choices:

Current Selection

The alert condition will be tested using whatever selections are made at the time of checking the alert.

All Clear

All selections will be temporarily cleared before testing the alert condition.

Bookmark

A specified bookmark will temporarily be applied before testing the alert condition.

Click **Next** to continue.

Step 3 - Define when to Test the Alert Condition

Step 3 - Define when to test the alert condition

Check Alert Interactively
In this step you define when the alert should be checked. The alert can be manually checked by means of Automation calls at any time. Automatic checks at relevant events can be made with the settings in this page.

On Open
Check alert interactively
Mark this check-box if the alert is meant to be checked automatically upon the events below.

On Post Reload
On Open
Mark this check box if the alert is to be automatically checked when the document is opened.

On Post Reduce
On Post Reload
Mark this check box if the alert is to be automatically checked after a Reload of the document (script execution).

Batch Alert
On Post Reduce
Mark this check box if the alert is to be automatically checked after a data reduction (Reduce data command) has taken place in the document.

Batch alert
Mark this check box if the alert is designed for triggering from external programs via the special Automation API for Batch triggers.

Click Next to continue.

< Back Next > Finish Cancel Help

Figure 46. The third page of the Alert wizard

In this step you define when the alert should be checked. The alert can be manually checked by means of Automation calls at any time. Automatic checks at relevant events can be made with the settings in this page.

Check Alert Interactively

Mark this check box if the alert is meant to be checked automatically upon the events below.

On Open

Mark this check box if the alert is to be automatically checked when the document is opened.

On Post Reload

Mark this check box if the alert is to be automatically checked after a **Reload** of the document (script execution).

On Post Reduce

Mark this check box if the alert is to be automatically checked after a data reduction (Reduce data command) has taken place in the document.

Batch Alert

Mark this check box if the alert is designed for triggering from external programs via the special Automation API for Batch triggers.

Click **Next** to continue.

Step 4 - Define test delay

Step 4 - Define test delay

Delay Days

In this step you may define a suspension period after the alert has been fired, in which the alert should not be checked. This makes it possible e.g. to make an alert fire only the first time you open a document each day.

Trigger Level

Delay
The re-firing of automatically checked alerts may be suspended for a given time after that an alert has been fired. Enter a number of days in this box. Decimal numbers may be used to specify part of a day.

Trigger Level
The firing of automatically checked alerts may also be suspended based on data changes. In this drop-down you can choose between three levels of reoccurrence.

Always
The alert is fired every time the trigger event(s) occur and the alert condition is fulfilled.

Message changes
The firing of the alert is suspended until the alert Message changes. This is of course only meaningful with dynamic message text. Note that the state of the alert condition does not necessarily have to change in order for the evaluated message to change.

State changes
The firing of the alert is suspended until the alert state changes, i.e. the alert condition has been unfulfilled during at least one check and becomes fulfilled again during a later check. This is the strongest type of alert suspension.

< Back Next > Finish Cancel Help

Figure 47. The fourth page of the Alert wizard

In this step you may define a suspension period after the alert has been fired, in which the alert should not be checked. This makes it possible e.g. to make an alert fire only the first time you open a document each day.

Delay

The re-firing of automatically checked alerts may be suspended for a given time after that an alert has been fired. Enter a number of days in this box. Decimal numbers may be used to specify part of a day.

Trigger Level

The firing of automatically checked alerts may also be suspended based on data changes. In this drop-down you can choose between three levels of reoccurrence.

Always

The alert is fired every time the trigger event(s) occur and the alert condition is fulfilled.

Message changes

The firing of the alert is suspended until the alert Message changes. This is of course only meaningful with dynamic message text. Note that the state of the alert condition does not necessarily have to change in order for the evaluated message to change.

State changes

The firing of the alert is suspended until the alert state changes, i.e. the alert condition has been unfulfilled during at least one check and the becomes fulfilled again during a later check. This is the strongest type of alert suspension.

Click **Next** to continue.

Step 5 - Define the action(s) to take when the alert fires

Step 5 - Define the action(s) to take when the alert fires

In this step you decide what actions should be taken when the alert is fired. If your alert is for check by macros only, this step may not be necessary.

Message
Here you type the message to be displayed with the alert. In case of pop-up alerts this is what appears in the pop-up. In case of mail alerts this is the body text of the mail. The text may be defined as a QlikView expression for dynamic update. By clicking on the ... button the full Edit Expression dialog is opened for easier editing of long formulas.

Show Popup

Send Mail

Subject
Mark this check box if you want the alert message to be shown as a pop-up balloon when the alert is fired.

Recipients
Mark this check box to send mail as a result of the alert being fired. The body of the mail will be the alert message text.

Mail Subject
Enter a text which will be used as mail subject in the sent mail(s). The subject may be defined as a QlikView expression.

Mail recipients
Enter a semi-colon separated list of mail addresses. Each of these addresses will receive an e-mail message whenever the alert fires. The list may be defined as a QlikView expression.

Click Finish to complete the creation of the new alert.

< Back Next > Finish Cancel Help

Figure 48. The fifth page of the Alert wizard

In this final step you decide what actions should be taken when the alert is fired. If your alert is for check by macros only, this step may not be necessary.

Message

Here you type the message to be displayed with the alert. In case of pop-up alerts this is what appears in the pop-up. In case of mail alerts this is the body text of the mail. The text may be defined as a QlikView expression for dynamic update. By clicking on the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Show Pop-up

Mark this check box if you want the alert message to be shown as a pop-up balloon when the alert is fired.

Send mail

Mark this check box to send mail as a result of the alert being fired. The body of the mail will be the alert message text.

Mail Subject

Enter a text which will be used as mail subject in the sent mail(s).
The subject may be defined as a QlikView expression.

Mail recipients

Enter a semi-colon separated list of mail addresses. Each of these addresses will receive an e-mail message whenever the alert fires.
The list may be defined as a QlikView expression.

Click **Finish** to complete the creation of the new alert.

16 VARIABLE OVERVIEW DIALOG

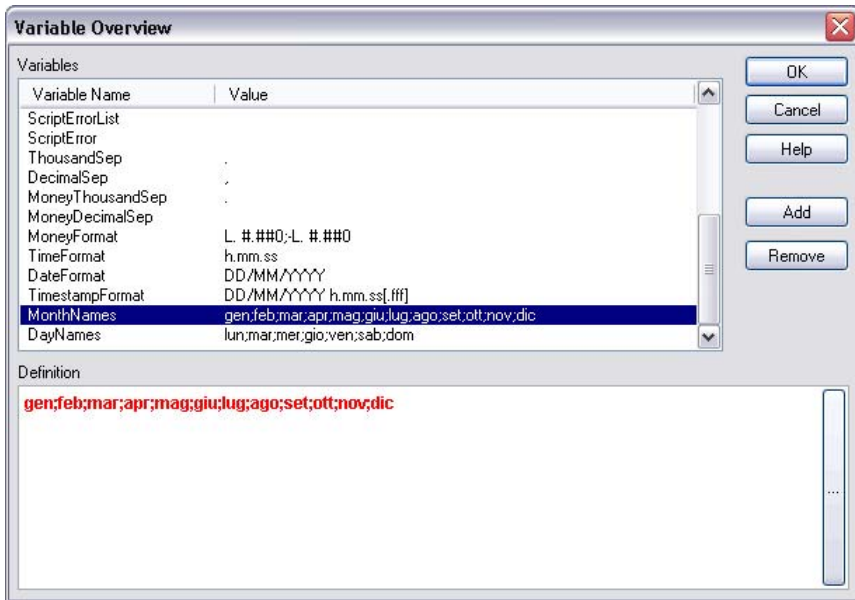


Figure 49. The Variable Overview dialog

This dialog is opened from the **Settings** menu and shows all non-hidden variables and their values in a single list. By clicking on a variable you may edit its contents in the lower edit box. Together with the dollar-sign expansion feature in QlikView expressions, the variable overview may be used as a crude expression repository.

Variables

In this two-column list you will find the names of all non-hidden variable and their respective values.

Definition

When clicking on a variable in the **Variables** list, the variable's contents will be shown in this edit box. You may then freely edit the text making up the variable value. The changes will be immediately reflected in the **Value** column of the **Variables** list. Click on the ... button in order to invoke the full **Edit Expression** dialog.

Add

Opens the **New Variable** dialog, where a new variable can be defined.

Remove

Removes the selected variable from the list.

17 EXPRESSION OVERVIEW DIALOG

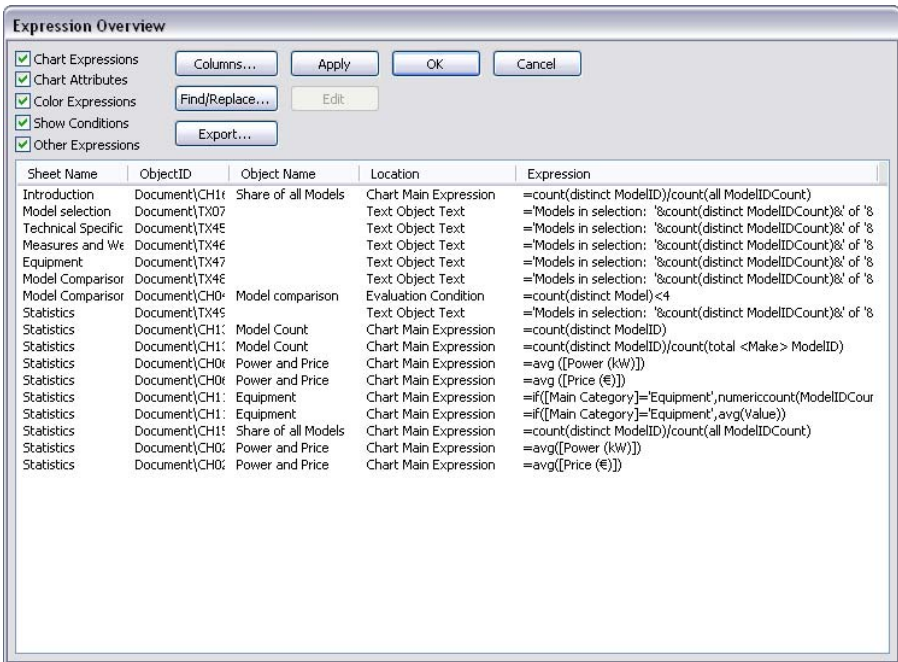


Figure 50. The Expression Overview dialog

This dialog is opened from the **Settings** menu and shows all document, sheet and sheet object expressions in a single list. From this list you may edit single expressions or do Find/Replace on multiple expressions. You can select one or more expressions from the list. A selected line is shown in black. Click below the last expression to deselect.

The expression list may contain the following columns, based on settings in the **Columns** dialog (see below):

Sheet ID

The sheet ID of the sheet where the expression occurs. Empty for document expressions.

Sheet Name

The name (tab title) of the sheet where the expression occurs. Empty for document expressions.

Object ID

The sheet object ID of the sheet object where the expression occurs. Empty for document and sheet expressions.

Object Name

The name of the object where the expression occurs (empty for document and sheet expressions).

Location

The location of the expression within properties with 'friendly name', i.e. a simple explanation of the type of expression.

Label

The label of the main expression of the object.

Location (Full)

The exact location of the expression within the properties structure. The notation corresponds to the object property hierarchies of the QlikView Automation Interface, which is explained in detail in the QlikView API guide.

Expression

The expression definition.

The following commands are available in the dialog:

Chart Expressions

Mark this check box to include chart expressions in the list.

Chart Attributes

Mark this check box to include chart attributes in the list.

Color Expressions

Mark this check box to include color expressions in the list.

Show Conditions

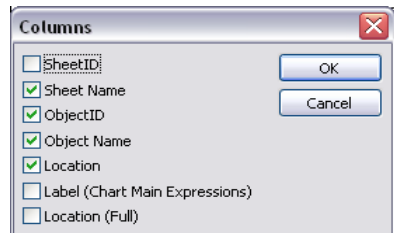
Mark this check box to include show condition expressions in the list.

Other Expressions

Mark this check box to include other types (than the three types listed above) of expressions in the list.

Columns...

Opens the **Columns** dialog where you may customize which columns that should appear in the **Expression**



Overview dialog. The **Expression** column cannot be turned off.

Edit...

After selecting a single expression in the list you may use this button to open the **Edit Expression** dialog for the selected expression.

Find/Replace...

Opens the **Find/Replace** dialog, where you may search for and replace specific text within the current list of expressions.

Export...

Use this button to export the table to a text file.

Apply

Applies changes made in the list back to the properties of the document, sheets and sheet objects, without leaving the dialog. After using this command changes made so far will remain even if you use **Cancel** to leave the dialog.

OK

Applies changes made and closes the dialog.

Cancel

Closes the dialog without applying changes (unless of course the **Apply** command has been used previously).

Changes made from this dialog can be undone by the regular **Undo** command.



18 INTERNAL FILES

18.1 File references in the QlikView layout

There are a number of places in the QlikView layout where external files can be referenced as info, as background images etc. External files are referenced by a local path or URL.

In certain cases it may be preferable to include the files in the QlikView document. This can be done as described under 18.2 below.

A small set of standard image files are embedded in the QlikView program itself and may be used in any document without any special preparations. This is described in the section 18.3 below.

18.2 Internal files in QlikView documents

The **info** prefix to **load** and **select** statements in the script (page 309 and page 329) can be preceded (or replaced) by the **bundle** prefix. The external files will then be read and embedded in the QlikView document. The external files will no longer be necessary for referencing the file contents, thereby making the QlikView document portable.

Show info and the info function will automatically use the embedded files when available. The syntax for explicit reference of embedded files is:

```
qmem://fieldnamefieldvalue |
qmem://fieldname<index>
```

where *index* is the internal index of a value within a field.

Examples:

```
'qmem://Country/Austria'
'qmem://MyField/34'
```

18.3 Internal files in the QlikView program file

A number of image files are available embedded in the QlikView program file. These may be used without any additional actions. The syntax for reference is:

```
qmem://<builtin>/filename
```

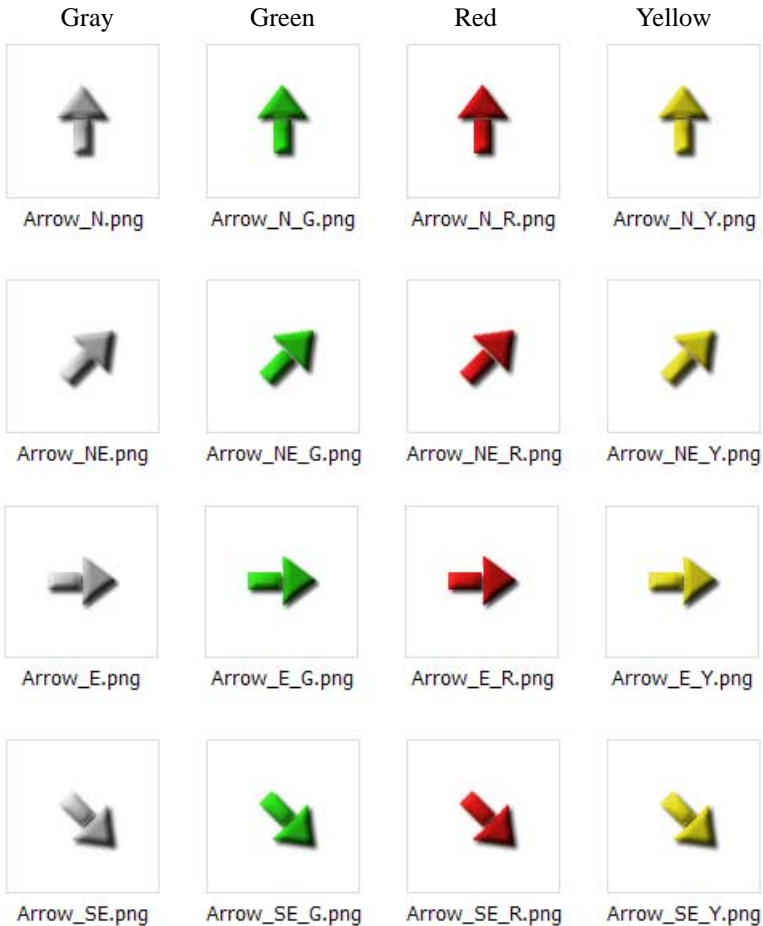
where *filename* is one of the filenames shown under the images listed below (including extension).

Examples:

```
'qmem://<builtin>/Arrow_N_G.png'  
='qmem://<builtin>/  
Smiley' & if (sum(Result) < 0, 1, 3) & '_Y.png'
```

Arrow images

Arrows are available in eight directions and four colors. Filenames are coded as follows: The word “Arrow” followed by an _ (underscore) and map direction. That may be followed by _G for green color, _R for Red color and _Y for yellow color. If no color is stated in the filename as above the color is gray.





Arrow_S.png



Arrow_S_G.png



Arrow_S_R.png



Arrow_S_Y.png



Arrow_SW.png



Arrow_SW_G.png



Arrow_SW_R.png



Arrow_SW_Y.png



Arrow_W.png



Arrow_W_G.png



Arrow_W_R.png



Arrow_W_Y.png



Arrow_NW.png



Arrow_NW_G.png



Arrow_NW_R.png



Arrow_NW_Y.png

Other images

A number of other images are also available. Colors are coded as for arrows. _B in the filename indicates blue color.



Check.png



Check_G.png



Cross.png



Cross_R.png



Minus.png



Minus_R.png



Plus.png



Plus_G.png



Smiley1.png



Smiley1_G.png



Smiley1_Y.png



Smiley2.png



Smiley2_B.png



Smiley2_Y.png



Smiley3.png



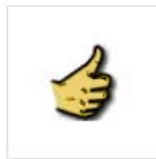
Smiley3_R.png



Smiley3_Y.png



Thumb1.png



Thumb2.png

18.4 Bundled Sounds

A number of sounds are available as bundled resources within QlikView. The sounds can be accessed via the following paths:

```
qmem://<bundled>/sounds/qv_btn.wav  
qmem://<bundled>/sounds/qv_can.wav  
qmem://<bundled>/sounds/qv_clear.wav  
qmem://<bundled>/sounds/qv_load.wav  
qmem://<bundled>/sounds/qv_lock.wav  
qmem://<bundled>/sounds/qv_menu.wav
```


qmem://<bundled>/sounds/qv_ok.wav
qmem://<bundled>/sounds/qv_redo.wav
qmem://<bundled>/sounds/qv_reex.wav
qmem://<bundled>/sounds/qv_save.wav
qmem://<bundled>/sounds/qv_scrpt.wav
qmem://<bundled>/sounds/qv_unlock.wav
qmem://<bundled>/sounds/qv_undo.wav

Note This list may change in future versions of QlikView.



PART IV: SCRIPT

- **Variables and Fields**
- **Script Dialog Boxes**
- **Script Syntax**
- **Data Structures**
- **Security**



19 VARIABLES AND FIELDS

This chapter explains the different data-carrying entities in QlikView, most importantly **Fields** and **Variables**.

19.1 Fields

Fields are the primary data-carrying entity in QlikView. *A field typically contains a number of values*, called field values. The QlikView internal logic works only on the fields and their field values.

Fields get their data in the script from **load**, **select** and **binary** statements reading database tables. The only way of changing data in a field (apart from input fields, see below) is by re-executing the script. *Field values can not be manipulated by the user* from the layout or from Automation. Once read into QlikView they can only be viewed and used for logical selections and calculations.

Field values can contain numeric or alphanumeric data. In case they are numeric they will always have dual values, the actual numeric value and the current (formatted) text representation of it. It is the latter which is shown in sheet objects etc.

19.2 Input Fields

Above it was stated that data in fields cannot be altered without running the script. No rule without an exception. Starting from version 8, QlikView supports a special type of field, which indeed can be changed without script execution, the input field.

An input field is just like any other field in QlikView, with the difference that its values, as read in the script, can later be changed without running the script again. Input fields can be used like any other fields in all types of sheet objects.

The script creates placeholders for each field value, which can later be edited to contain new data. Any field can be turned into an input field by listing it in an **inputfield** statement in the script before it appears in a **load** or **select** statement.

Values of an input field can be changed in list box cells, table box cells and in table chart expression cells. Only list boxes and table columns which contain input fields are editable. By hovering over an editable cell you will see an input icon. Clicking the icon sets the cell in input edit mode. It is possible to use up/down arrow keys to move between cells while staying in input edit mode. The entire QlikView document will automatically recalculate whenever new values are entered.

Note Dynamic update cannot be used on input fields. The input field functionality will automatically be disabled.

A table chart expression cell may contain the aggregation **inputsum** (*inputfield*) and still be open for input. The change will then be distributed back to the underlying field values, based on predefined algorithms, such as e.g. “spread equally“ or “spread proportionally“.

Version 9 introduces an extension to the input fields, namely the possibility to specify relative change. The following syntax applies (*n* is a number):

%+n increases the current value by n%
%-n decreases the current value by n%
+ =n increases the current value by n
- =n decreases the current value by n
*** =n** multiplies the current value with n
/ =n divides the current value with n.

Example:

%+10 increases the current value with 10%.
+ =56 increases the current value with 56.
*** =2** multiplies the value with 2.
/ =2 divides the value with 2.
/ =0 no change.

There are also Automation APIs for extracting and setting values programmatically.

19.3 Variables

Variables in QlikView are named entities containing a *single data value*.

Variables can get their values from **LET**, **SET** or other control statements in the script. However they can also be created and receive values via Automation calls or by the help of input boxes in the layout. The value of a variable can normally be changed by the user at any time.

Variables can contain numeric or alphanumeric data. If the first character of a variable value is an equal sign '=' QlikView will try to evaluate the value as a formula (QlikView expression) and then display or return the result rather than the actual formula text.

19.4 Other Entities in QlikView

Chart Dimensions

A chart dimension is the set of values for the chart to iterate over when it calculates the values for its expression(s). In the simple case one could say that it is what appears on the x-axis in a standard bar chart.

A chart can have one or more dimensions. The upper limit depends on chart type, data complexity and the amount of memory available. Pie, line and scatter charts can show a maximum of two dimensions, bar and grid charts three. Radar charts can only show one dimension, gauge charts show no dimensions at all.

A chart dimension gets its values from a field which is specified on the **Dimensions** page of the **Chart Properties** dialog. Instead of being a single field a dimension can also consist of a group of fields (see **Groups** below).

Chart dimensions may also be calculated from an expression.

Groups

Groups, or more correctly field groups, are collections of fields, which can be used instead of fields in chart dimensions. There are two types of field groups:

Drill-down groups are used to create hierarchies of fields allowing so called drill-down in charts.

Cyclic groups are non-hierarchical and are normally used just as a convenient way of letting the user switch chart dimension fields by a simple mouse click.

Groups are specified on the **Groups** page of the **Document Properties** dialog.

Expressions (Calculated Formula)

An expression consists of a combination of fields, variables, operators and functions put together according to a special syntax. The functions available vary slightly between script, charts and other parts of the layout.

In the script the expression will be evaluated as the script execution passes it by. In charts and other layout objects expressions will be evaluated automatically whenever any of the fields, variables or functions that the expression contains changes value or logical status.

For more information about script expressions see chapter “Script Expressions” on page 355.

For more information about chart expressions see chapter “Chart Expressions” on page 293 in Book III.

For more information about the use of expressions in other layout objects see chapter “Calculated Formulas” on page 467 in Book III.

20 SCRIPT DIALOG BOXES

20.1 Edit Script

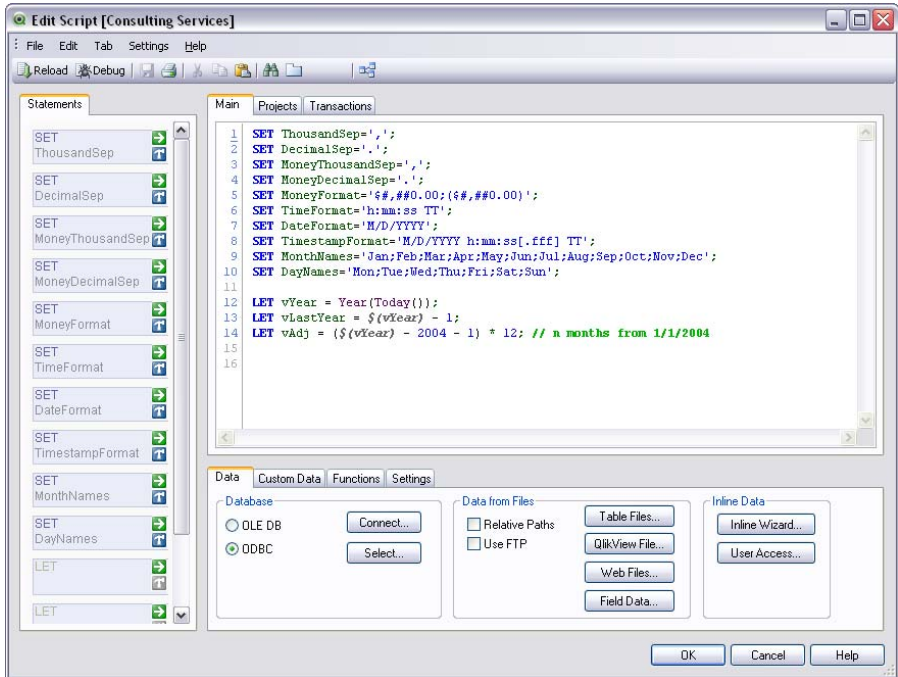


Figure 51. The Edit Script dialog

The **Edit Script** dialog box (above) is opened from the **File** menu or with a button in the toolbar. Here you can enter and execute a script that connects your application to an ODBC data source, or to data files of different types, and fetches the demanded information.



Scripts can be typed manually, or generated automatically. Complex script statements must, at least partially, be entered manually. The statements, expressions, functions etc. available for the creation of scripts are described in the chapters “Script Syntax” on page 281 and “Script Expressions” on page 355.

The **Edit Script** dialog window is sizable (point at the lower right-hand corner with the mouse, then drag).

The dialog contains a number of different components:

At the top you can find a **menu bar** with various script related commands. The most frequently used commands are also found on the adjacent **toolbar**.

The **script pane** to the upper right contains the actual script. This pane may be divided into different tabs.

The **statement graph** to the left is used for easier navigation in the script. Each statement is represented by a box outlining the most important properties of the statement.

At the bottom of the dialog you will find a **tool pane** with multiple tabs containing functions for script generation.

The dialog contains the following buttons:

OK

Accepts changes made in the script and closes the dialog box.

Cancel

Cancels the script editing, restores the script and closes the dialog box.

Help

Invokes online help.

Menus and toolbar

File menu

Reload

Executes the script, closes the dialog box and opens the **Fields** page in the **Sheet Properties** dialog box. This command is also available on the Edit Script dialog toolbar.



Debug...

Starts the script execution in the debugger. The debugger is used for searching for errors in the script. Every script statement can be monitored and the values of the variables can be examined while the script is executed. See page 277. This command is also available on the **Edit Script** dialog toolbar.



Edit Hidden Script...

Opens a script tab for hidden script. Here it is possible to define a hidden part of the script, which is executed before the general script at each reload. Hidden script tabs are always shown to the left of the open script tabs. A password is required to create or access hidden script. See page 227 for details.

Change Hidden Script Password...

Opens the **New Hidden Script Password** dialog where a new password for access to the hidden script can be set.

Create Hidden Script

Opens the **New Hidden Script Password** dialog, where a password for the hidden script must be set, and then opens a new script tab in the script window called **Hidden Script**. Here it is possible to define a hidden part of the script, which is executed before the general script at each reload. Hidden script tabs are always shown to the left of the open script tabs.

Remove Hidden Script

Removes the hidden script from the **Edit Script** dialog.

Save Entire Document

Saves the active document in a file. Data, script and layout are saved. This command can also be invoked via the keyboard shortcut CTRL+S. This command is also available on the **Edit Script** dialog toolbar.

**Save Entire Document As...**

Saves the active document in a new document file under a new name.

Export to Script File...

Saves the entire script in a text file to be specified in the **Save script** dialog that appears. The file will have the extension .qvs.

Print Tab...

Allows you to print the active script tab on a printer of your choice. This command can also be invoked via the keyboard shortcut CTRL+P. This command is also available on the **Edit Script** dialog toolbar.

**Print All Tabs...**

Allows you to print the entire script on a printer of your choice. Hidden script will not be printed via this command.

Table Viewer

Opens the **Table Viewer** dialog where the structure of the loaded data can be studied in a graphical view of tables, fields and associations. This command can also be invoked via the keyboard shortcut CTRL+T. Read more about the **Table Viewer** on page 228.

**Edit menu****Undo**

Undoes the latest change (multiple step undo is possible). Equivalent to pressing CTRL+Z.

Redo

Redoes the latest **Undo**. Equivalent to pressing CTRL+Y.

Cut

Exports the selected text to the Clipboard. This command is also available on the **Edit Script** dialog toolbar. Equivalent to pressing CTRL+X.

**Copy**

Copies the selected text to the Clipboard. This command is also available on the **Edit Script** dialog toolbar. Equivalent to pressing CTRL+C.

**Paste**

Pastes the contents of the Clipboard into the dialog at the position of the cursor. This command is also available on the **Edit Script** dialog toolbar. Equivalent to pressing CTRL+V.

**Delete**

Deletes the text in the script that has been selected. Equivalent to pressing DEL.

Clear Entire Script

Clears the script, i.e. removes all the text except the initial autogenerated **set** statements.

Select All

Selects the entire script text. Equivalent to pressing CTRL+A.

Insert File...

Lets you browse for a file containing a script or part of a script and inserts its contents into the script at the position of the cursor. If the **Internet** check box is marked, see further on page 239.

Include

Opens the **Include Scriptfiles** dialog. Script files can have one of the following formats: .qvs (a previously saved QlikView script file, .txt or .sql).

Insert Domain SID

Retrieves the NT domain security ID for use in the script. The NTDOMAINSID is employed in the **section access** as one of the reserved fields that govern NT security.

Find/Replace...

Opens a dialog allowing you to find and replace numbers or characters in the script. This command is also available on the **Edit Script** dialog toolbar. Equivalent to pressing CTRL+F.



Goto...

Opens a dialog allowing you to go to a specified line number in the script. Equivalent to pressing CTRL+G.

Upper Case

Puts the selected script text in upper case.

Lower Case

Puts the selected script text in lower case.

Invert Case

Inverts the casing of the selected script text.

Capitalize

Capitalizes the selected script text.

Comment

Transforms a text row in the script to a comment.

Uncomment

Transforms the text row back into original script text.

Tab Menu

Add Tab

Adds a new script tab. The script will be executed tab by tab in order from left to right. This command is also available on the **Edit Script** dialog toolbar. If the current script tab is part of the hidden script, the new tab will also be created within the hidden script.

Insert Tab at Cursor

Inserts a new tab after the active tab. Any text on the active tab positioned after the cursor will be moved to the new tab.

Rename...

Opens a dialog for renaming the active script tab.

Promote

Moves the active tab one step to the left. Tabs cannot be promoted to the left of the hidden script tab.

Demote

Moves the active tab one step to the right.

Merge with Previous

All text on the active tab is moved to the end of the preceding tab and the active tab is deleted.

Remove...

Removes the active script tab. The last remaining script tab cannot be removed.

Settings menu

Configure

Opens the **Editor** page in the **User Preferences** dialog (page 99), where you can set the font and color of the different text types appearing in the script.

Help menu

Help

Opens the QlikView help program.

Script Pane

The script pane contains the actual script. Each line of the script is numbered. The script may be divided into several parts on separate tabs. The tabs are executed from left to right.


If a hidden script is used it may be viewed as a separate tab to the very left (provided you know the password).


The script is color coded by syntax components. The color coding may be customized by choosing **Configure** on the **Settings** menu.

Statement Graph

The statement graph shows a box for each statement on the active script tab. The box outlines the most important features of the statement and provides an easier way to navigate the script.



By clicking the  icon you will move the cursor in the script pane to the beginning of that statement's code.

By clicking the  icon you will re-enter the wizard for the statement. This option is only available for certain statements. For more complex statements, it is sometimes possible to reflect the statement in the wizard context.

Tool Pane

The tool pane has four tab pages containing functions for script generation.

Data Page

The **Data** page contains basic commands for getting data into QlikView.

Database

The commands in the **Database** group are used to create a connection to and select fields from a data source. If you are using a commercial DBMS you may use ODBC or OLE DB as an interface

between QlikView and the database. An ODBC or OLEDB driver that supports your DBMS must be installed on your computer before you can retrieve data over the ODBC/OLE DB interface. Once the appropriate ODBC/OLE DB driver is installed, your database must be configured as an ODBC data source.

OLE DB

Mark this alternative to access databases through OLE DB.

ODBC

Mark this alternative if you wish to access databases through ODBC.

Connect...

Opens the **Data Link Properties** dialog box (Figure 54 on page 231) where an ODBC or an OLE DB data source can be selected. Pressing **OK** will generate an appropriate **connect** statement.

Select...

Opens the **Create Select Statement** dialog box (Figure 56 on page 233) in which you can specify fields and tables from the chosen data source.

Data from Files

The commands in the **Data from Files** group are used for reading data from files.

Relative Paths

If this check box is marked, QlikView will use relative instead of absolute paths in the script. Relative paths are normally required when an application is to be moved between different computers. This setting is also available in the **User Preferences** dialog. If it is changed there, it will also be changed in the **Edit Script** and **Edit Hidden Script** dialog boxes.

Use FTP

If this check box is marked, a dialog allowing you to select files from an ftp file server will open when you click **Table Files**, **QlikView Files**, or **Include**. See Figure 58 on page 239.

Table Files...

Opens the **Open Local Files** dialog as part of the **File Wizard** and shows a listing of table files. Selecting one or several files and pressing **OK** will generate one or several **load** statements. When executed, the script loads data from the

corresponding files. If the **Use FTP** option is checked (see above), select a server from the list (or enter the server you wish to use, then click **Connect**). When this is done, select a file.

QlikView File...

Opens the **Open QlikView File** dialog box listing QlikView files (.qvw). Selecting a file and pressing **OK** will generate a **binary** statement. When executed, the script loads data from the corresponding file, however without using the layout settings. If the **Use FTP** option is check marked (see above), select a server from the list (or enter the server you wish to use, then click **Connect**). When this is done, select a QlikView file.

Web Files...

Opens the **File Wizard** on the **Source** page where you may enter a URL as a source for your database table.

Field Data

Opens the **File Wizard** on the **Source** page where you may load the contents of an already loaded field.

Inline Data

The options in the **Inline Data** group are used for generating data inline in the script.

Inline Wizard...

Opens the **Inline Data Wizard** with the help of which you can create load inline statements from a spreadsheet style control. See page 241 for detailed information about this dialog.

User Access...

Opens the **Access Restriction Table Wizard** with the help of which you can create load inline statements for **section access** in the script. See page 244 for detailed information about this dialog.

Custom Data Page

The **Custom Data** page contains commands for getting data from custom data sources into QlikView.

Custom Data Sources

QlikView offers an open source plug-in interface, providing possibility to program custom interfaces to various types of data sources not covered from the traditional file, ODBC or OLEDB interfaces.

The typical case is data available via Web Services. The plug-in should be programmed according to specifications shown in a template code provided (on request) as open source from QlikTech and compiled as a dll. The dll is then placed next to the QV.EXE file making the custom source available to use. It will then appear in the drop-down box for selection.

Connect...

Opens the **Connect** dialog for connecting to the custom data source. This dialog may look different depending on the what data source is used.

Select...

Opens the **Select** dialog box for selecting fields from the custom data source. This dialog may look different depending on what data source is used.

Functions Page

The **Functions** page contains tools for navigating and pasting data QlikView functions.

Function Category

This drop-down box contains a list of function categories. Select a category in the list to see the corresponding functions in the **Function Name** list below.

Function Name

This drop-down box contains a list of QlikView standard script functions. The list can be narrowed down by selecting a category in the **Function Category** list above.

Paste

Select a function in the **Function Name** list above and press this button to paste it at the cursor position in the script.

Settings Page

The **Settings** page contains security settings for the QlikView script.

Script Privileges**Open Databases in Read and Write Mode**

If this option is selected the **mode is write** qualifier in **select** statements will be enabled for use.

Can Execute External Programs

If this option is selected **execute** statements may be used in the script.

Settings

Scramble Connect User Credentials

If this check box is marked, the USERID and PASSWORD in **connect** statements will be scrambled in the script.

The Find/Replace Dialog

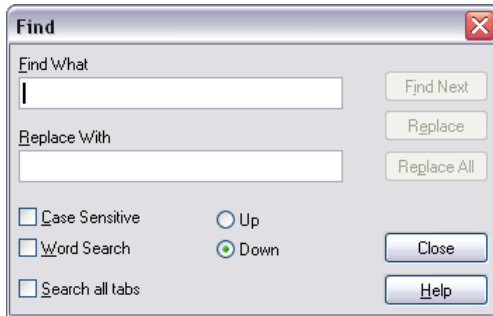


Figure 52. The Find/Replace dialog

This dialog is used for searching for specific text strings within the current list of expressions and for making multiple or bulk changes to the expressions. It opens from the **Edit** menu.

Find What

The text string to search for.

Replace With

The text to replace the search string with.

Find Next

Moves selection to the next expression containing the search string.

Replace

Makes a replacement in the selected expression.

Replace All

Makes replacements in all occurrences of the search string.

Case Sensitive

If this check box is marked the text search will be case sensitive.

Word Search

If this check box is marked QlikView will only find occurrences of the search string forming a whole word (delimited by spaces or other non-letter characters).

Search All Tabs

If this check box is marked, the find/replace operation will be made over all sheet tabs.

Up

Mark this radio button to search up through the expression list.

Down

Mark this radio button to search down through the expression list.

Close

Closes the dialog.

Help

Opens the online help.

20.2 Hidden Script

A hidden script is a closed part of the script, which will be executed before the normal script at each **Reload**. The hidden script is protected by a password.

When choosing **Edit Hidden Script** from the **File** menu in the **Edit Script** dialog you will be prompted for a password, which will be required before giving access to the hidden script again. If it is the first time you access a hidden script in a document (thereby creating one) you will have to confirm the new password. After this the Hidden Script tab will appear to the left of all other script tabs and remain there until you close the document.

Note If a hidden script is used, the **binary** command cannot be used in the normal script.

Note The **Progress** dialog will not be updated during the execution of a hidden script, unless **Show Progress for Hidden Script** is marked in **Document Properties, Security**. No entries will be made in the log file, if used.

Note If the hidden script contains a **section access**, such a section will not be permitted in the normal script or in a script starting with a binary load of the QlikView file containing the hidden script.

20.3 Table Viewer

The **Table Viewer** dialog is opened when you choose **Table Viewer** from the **File** menu or press CTRL+T.

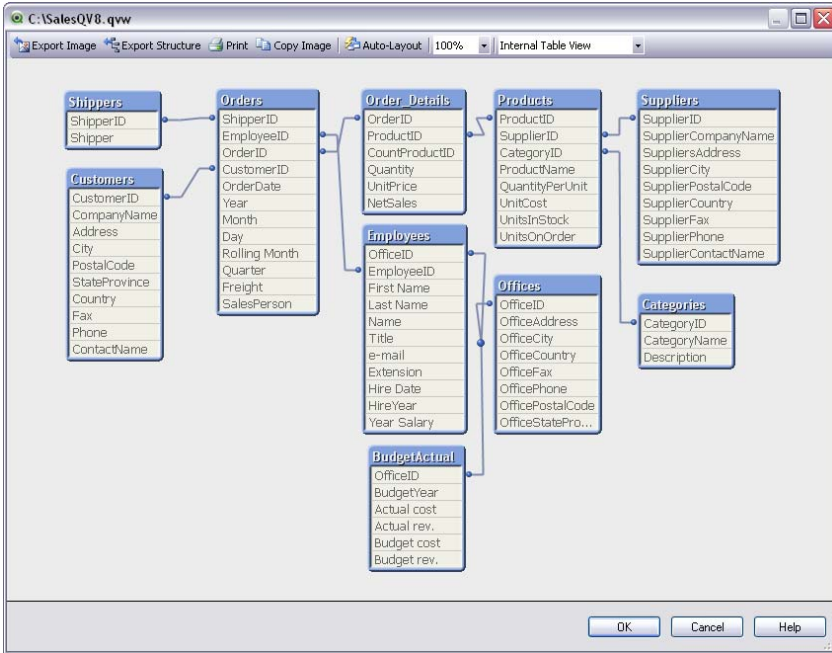


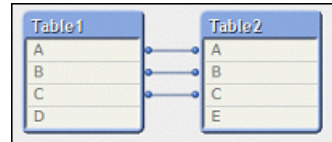
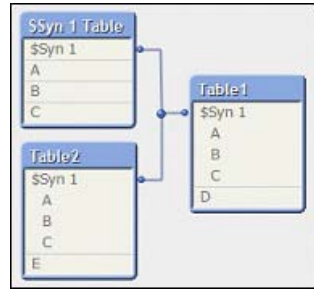
Figure 53. The Table Viewer dialog

This dialog is used to display the data table structure of the current QlikView document. Tables are shown as boxes with a list of the fields they contain. Between the boxes there are connector lines showing the associations. Where more than two lines meet there are connector points in the form of small dots.

The tables, the connector points and also all points where the connectors bend can be moved by means of mouse drag and drop (see below). When you re-arrange the layout like this, your changes will be saved with the document unless you exit the dialog with **Cancel**.

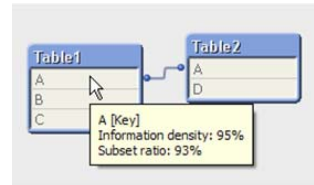
There are two views available, the layouts of which are stored separately. The two views are:

- The **Internal Table View** shows the data tables as QlikView stores them. Composite synthetic keys are formed in tables that share more than one field. Synthetic tables are used to link them. This view offers the best understanding of the QlikView logic and also provides a very clean layout where each pair of tables has a maximum of one connector between them. This is the default view.
- The **Source Table View** shows data tables as QlikView reads them. Here you have no synthetic fields or synthetic tables. Composite keys are represented by multiple connectors between tables.



When hovering with the cursor above one of the fields in a table, a tool tip pops up with information about the content of this field:

- **Information density** is the number of records that have values (i.e. not NULL) in this field as compared to the total number of records in the table.
- **Subset ratio** is the number of distinct values of this field found in this table as compared to the total number of distinct values of this field (i.e. also other tables). The following actions can be performed in the table layouts:



Click on Table Caption

Highlights the table, all tables with direct logical associations with that table and the connections between them.

Point at Table Caption and Drag

Moves the table in the layout.

Hover Over Table Caption

Shows pop-up info about table.

Click on Field in Table

Highlights the table, the field, all tables containing that field and the connections between them.

Hover over Field in Table

Shows pop-up info about field.

Right-click in a Table

Click on **Preview** to see a dialog of sample records from the loaded data.

TMP_KEY_FOR...	TMP_KEY_FOR...	Name_2
ine	ine	Indo-European
mol	mol	Moldavian
mol	mol	Moldavian
nuo	nuo	Nuoresiska
nuo	nuo	Nuoresiska
put	put	Puter
put	put	Puter

Point at Connector Point or Connector Bend and Drag

Moves the connector point in the layout.

Right-click on Connector Point or Connector Bend

Moves the connector point back to automatic positioning.

Click on Layout Background

Cancels all highlighting of tables, fields and connectors.

The following commands are available in the dialog toolbar and buttons:

Export Image

Opens a file browser dialog so that you can save the current table view as a bitmap or png image to a file on disc.



Export Structure

By pressing this button you can export the table structure of the document to a set of text files. These text files, one for the tables (*filename.Tables.tab*), one for the fields (*filename.Fields.tab*) and one for mapping in between (*filename.Mappings.tab*) can easily be read back into QlikView for further analysis with the full power of the QlikView logic. A dialog will appear where you can choose the target folder for the export. The default is to put the files in the same folder as the QlikView document.



Print

Opens the **Print** dialog from which you can print the current table view. Equivalent to pressing CTRL+P.



Copy Image

Copies the current table view as an image to clipboard. Equivalent to pressing CTRL+C.



Auto Layout

Re-arranges the tables in the current view.



Zoom

The zoom factor of the current view can be set in this drop-down box.

View

Select between **Internal Table View** and **Source Table View**.

OK

Saves table viewer layouts and closes the dialog.

Cancel

Closes the dialog without saving changes in the table viewer layouts.

20.4 Data Link Properties

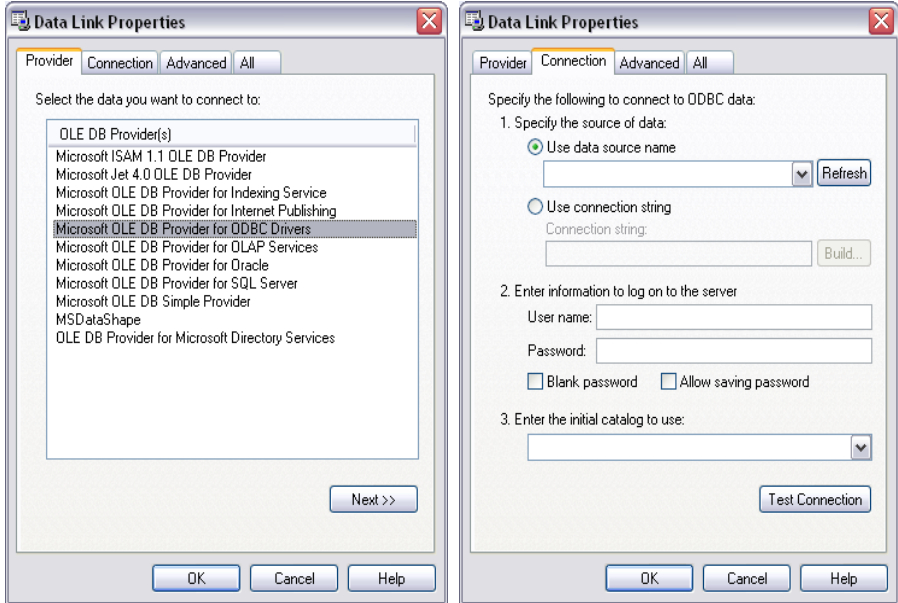


Figure 54. *Data Link Properties dialog*

This dialog box is used for creating a connection to an OLE DB data source, typically via ODBC. The dialog is opened when you click the **Connect...** button in the **Data** page of the **Tool Pane** in the **Edit Script** dialog box.

The first page lists the names of the OLE DB providers. In order to create an ODBC data connection, select *OLE DB Provider for ODBC Drivers*, then click **Next>>** to get to the **Connection** page.

On the **Connection** page under **1. Specify the source of data**, you can specify which previously configured data source you wish to use. There are two different types of ODBC data sources that may come into question.

Pre-Defined Data Sources

Data sources which have been predefined according to the procedure described on page 41 can be accessed by selecting the radio button **Use data source name**. The drop-down list contains all the data sources that are defined. Select the data source you wish to use.

You may if necessary specify **User name** and **Password** for the data source. You can also test the connection by pressing the **Test Connection** button.

Finally click the **OK** button. Your **connect** statement is ready.

Generic Data Sources

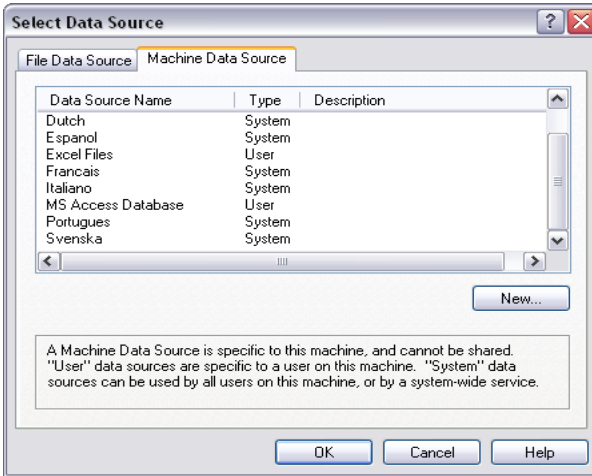


Figure 55. The Machine Data Source page of the Select Data Source dialog

Generic data sources are used e.g. for database files. If you want to use a generic data source or create a new source, select the **Use Connection String** radio button and click **Build**. This opens the **Select Data Source** dialog.

There are two types of data sources available: File data sources and Machine data sources. The latter data sources are specific to the local machine and user while file data sources are specific to a file. You may use any of the two. Each has a separate page in the **Select Data Source** dialog.

Double-click a data source, then browse to the applicable database file in the dialog that opens.

Note For MS Access data sources you must click the **Database** button in an intermediate dialog.

When you hit RETURN you will be returned to the second page of the **Data Link Properties** dialog.

You may if necessary specify **User name** and **Password** for the data source. You can also test the connection by pressing the **Test Connection** button.

Finally click the **OK** button. Your **connect** statement is ready.

20.5 Create Select Statement

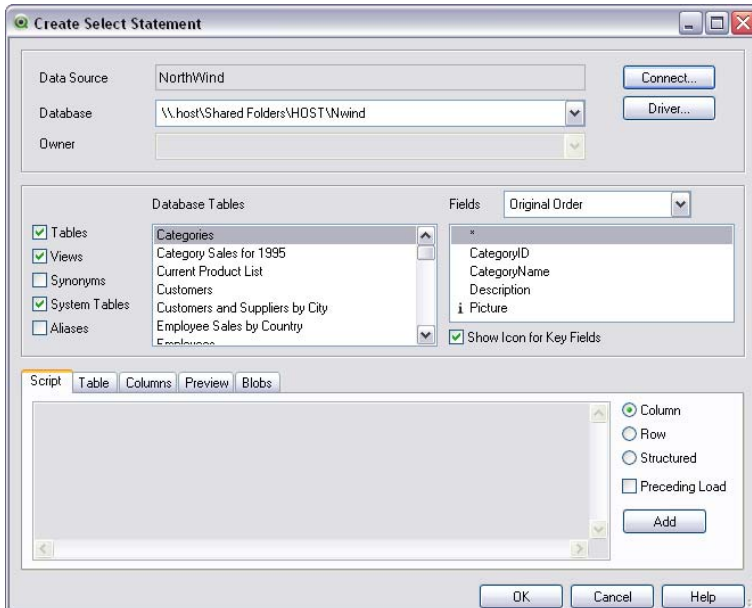


Figure 56. The Create Select Statement dialog

This dialog box is opened by clicking the **Select...** button in the **Data** page of the **Tool Pane** in the **Edit Script** dialog box. It is used to define the tables and fields to be loaded from previously selected data sources. The dialog can be sized by dragging the lower right corner. It consists of three panes.

The top pane contains information about the data source.

Data Source

Here the name of the current ODBC/OLEDB data base is shown.

Database

Here the name of the database within the defined data source is shown. If more than one database is available within the data source, you may choose between available databases via a drop-down list.

Owner

Here the name of the database owner is shown. If more than one owner is available, you may choose between available owners via the drop-down list.

Connect...

By clicking this button, you get to the **Data Link Properties dialog** (see above, Figure 54 on page 231), where you choose a predefined data source or create a new one.

Driver...

Click this button to open the **ODBC Driver Information** or **OLEDB Driver Information** dialog, where you can see information about supported functionality in the driver currently used. The information available may vary between drivers, but typically includes SQL functionality supported and limits.

The middle pane is used for selecting tables and fields.

Database Tables

This list shows all available database tables. Click a table name to select it. The types of tables that will appear in the list can be controlled via the check boxes to the left (see description below).

Fields

This list shows all available fields in the selected table. Choose one or more fields from the list to be included in the select statement. You can use "*" in order to select all of the field names at once. Your selection of fields will be reflected in the **Script** page of the bottom pane (see below). Fields can be sorted in **Original Order** from the database or in alphabetical **Text Order** by selection in the drop-down control.

Show Icon for Key Fields

If this check box is marked, all fields that are defined as keys in the source database will be shown with a key icon in the **Fields** list.

Tables

This check box must be selected if regular database tables are to appear in the **Database Tables** list.

Views

This check box must be selected if database views are to appear in the **Database Tables** list.

Synonyms

This check box must be selected if database table synonyms are to appear in the **Database Tables** list.

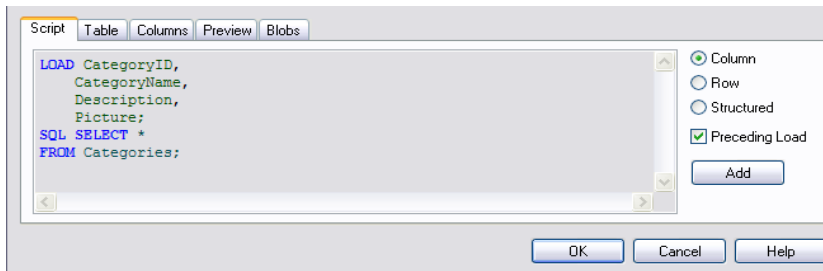
System Tables

This check box must be selected if database system tables are to appear in the **Database Tables** list.

Aliases

This check box must be selected if database aliases are to appear in the **Database Tables** list.

The bottom pane shows the generated select statement and information about the selected table and fields. The pane contains the following pages.

**Script**

The script page shows the script about to be generated for the **select** statement. The three radio buttons to the right in the bottom pane are used to control the formatting of the select statement in the script.

Column

Select this option to generate the **select** statement with each field name appearing on a separate row in the script.

Row

Select this option to generate the entire **select** statement appearing on a single row in the script.

Structured

Select this option to generate the **select** statement on structured rows in the script.

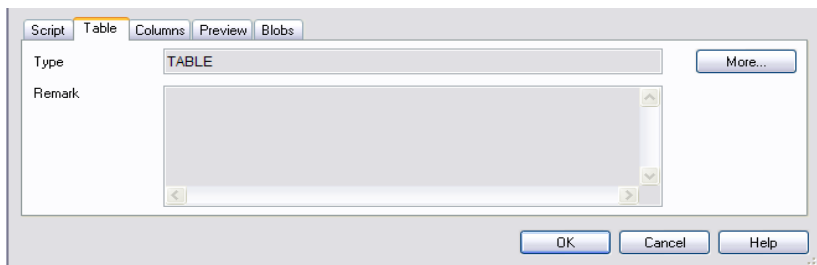
Preceding Load

If this check box is marked the **select** statement will be generated with a preceding **load** statement using the select

statement as input. All fields will be listed in the load statement even if * is used in the select statement.

Add

When clicking the **Add** button the select statement generated so far will be saved and the generation of a new select statement may begin without leaving the dialog. The old statement will be visible in the preview pane above a clear divider. This procedure may be repeated any number of times.



Table

The **Table** page shows detailed information about the selected database table.

More...

Opens a separate sizable dialog with an expanded view of the **Table**, **Columns** and **Preview** pages described above.


Columns

The **Columns** page shows detailed information about the selected columns (fields) of the current table. The **More...** control described under **Table** above is also available on this page.

Preview

The **Preview** page shows a preview of the first rows that will be generated by the current **select** statement. The **More...** control described under **Table** above is also available on this page.

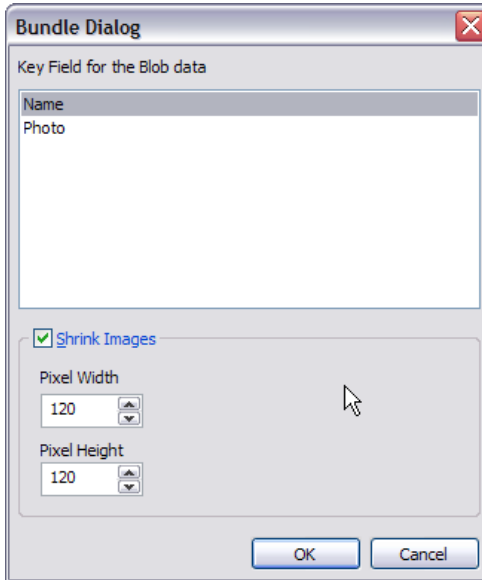
Blobs

The **Blobs** page makes it possible to bundle so-called Blobs, Binary Large Objects, in a QlikView application. It is usually a picture, a text file or similar, stored as a single entity in a database management system. The  icon indicates that the field contains a blob. The blobs in the database are listed together with size, in KB, and

type in the **Blob Viewer**. The Blob is previewed next to the list if it is an image. QlikView supports jpg, png, bmp, pdf and rtf blobs.

Bundle

The **Bundle** button opens the **Bundle Dialog** where the **Key Field for the Blob data** is selected and the blob can be resized to fit in the table cell if it is an image. Mark **Shrink Images** and apply width and height to resize the images to fit. The setting does not handle aspect ratio.



Note It is only possible to bundle Blobs using ODBC!

Once you have made your selections of table and fields you may click **OK** to insert the **select** statement generated at the cursor position in the QlikView script. Pressing **Cancel** will abandon changes.

The ODBC driver can usually interpret far more complicated **select** statements than the ones which can be produced by this dialog. An alternative way to generate a more complex **select** statement is to use a query tool, e.g. Microsoft Query, and graphically make your **select** statement there. When this is done, copy (CTRL+C) the **select** statement and paste it (CTRL+V) into the QlikView script. In MS Query, press the SQL button.

20.6 Open QlikView File

This dialog box is opened by clicking the **QlikView File...** button in the **Data** page of the **Tool Pane** in the **Edit Script** dialog box (Figure 51 on page 217).

In the **Open QlikView File** dialog box, which looks very similar to the **Open Local Files** dialog (Figure 57 below), you can specify the qvw file to load data from. A multiple file selection is not possible in this dialog since only one binary file can be loaded. When the **Open** button is clicked, a **binary** statement that loads the selected file will be generated in the script.

If the **Use FTP** option in the **Edit Script** dialog is selected, this dialog looks somewhat different. See Figure 58 on page 239.

20.7 Open Local Files

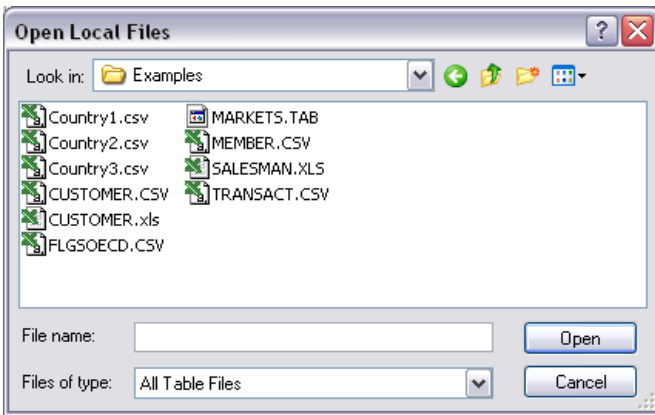


Figure 57. The *Open Local Files* dialog

The **Open Local Files** dialog box (Figure 57) is opened by clicking the **Table Files...** button in the **Data** page of the **Tool Pane** in the **Edit Script** dialog box (Figure 51 on page 217).

In the **Open Local Files** dialog you can specify the table files to study. In the drop-down list of file types, it is possible to specify the file type and delimiter in the file. When the **Open** button is clicked, a **load** statement that loads the selected files will be generated in the **Edit Script** dialog box. For multiple file selections, use CTRL-click or SHIFT-click.

If the **Internet** option in the **Edit Script** dialog is selected, this dialog looks like the one below (Figure 58).

20.8 Open Internet Files

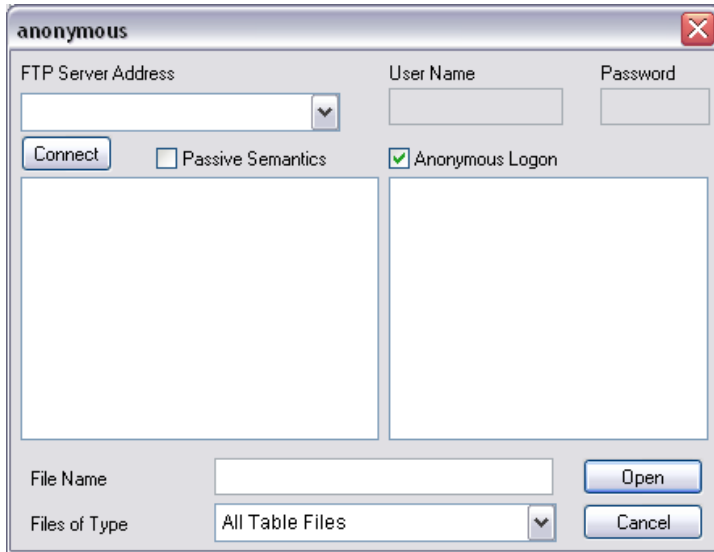


Figure 58. The Open Internet files dialog

This dialog box, allowing you to select one or several files from an Internet source and read them into QlikView by means of the script, appears when you click one of the buttons **QlikView File...** or **Table Files...** in the **Data** page of the **Tool Pane** in the **Edit Script** dialog - provided that the **Use FTP** check box is marked. Depending on whether you open it from **QlikView File...** or **Table Files...** the dialog is called **Open QlikView Document** or **Open Internet Files**, but the functionality is the same. You can also use this dialog to open QlikView documents. This is done by choosing **Open Ftp** from the **File** menu.

Select the server containing the file(s) you wish to open in the drop-down list box. If your server is not in the list (or if there is no list), enter a server name, then click **Connect**.

User Name

If the **Anonymous Logon** check box is not marked, enter your user name here.

Password

If the **Anonymous Logon** check box is not marked, enter your password here.

Connect

Click this button to connect to the server. If you select a server from the list, you don't need to click this button.

Passive Semantics

Mark this checkbox if the connection should utilize passive ftp.

Anonymous Logon

If this box is checked, you can log on without specifying your user name and your password. If the server requires this information, an error message will appear.

Once connected to the ftp server, browse through the files to find the one you want to use.

File Name

The name(s) of the selected file(s) will appear in this box.

Files of Type

Here you can specify the type and delimiter of the file you want to open.

When the **Open** button is clicked, a **load** statement that loads the selected files will be generated in the **Edit Script** dialog box. For multiple file selections, use CTRL-click or SHIFT-click.

If this dialog was opened from the **File** menu, clicking the **Open** button means opening the file in QlikView.

20.9 Inline Data Wizard

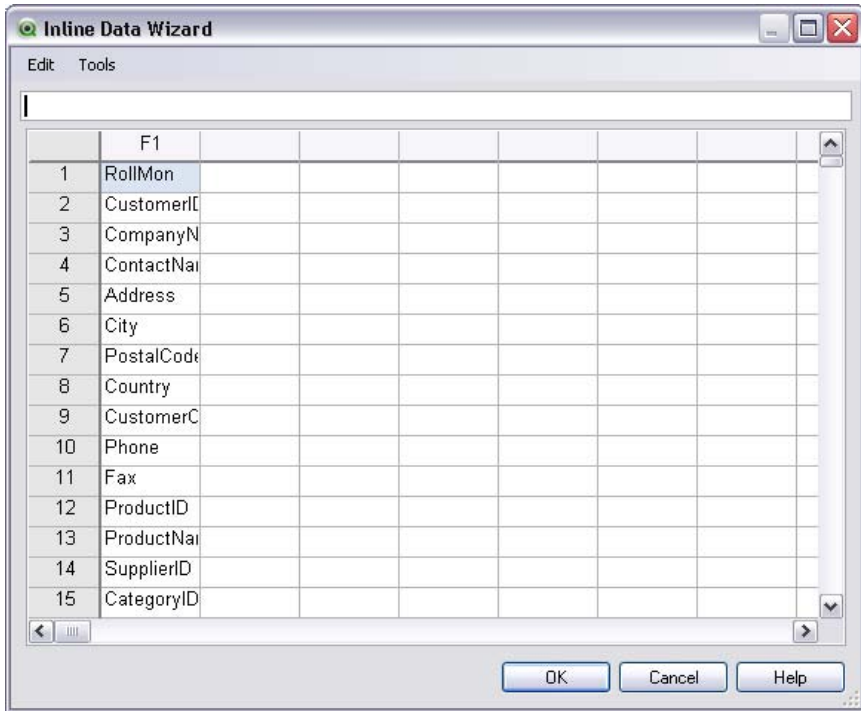


Figure 59. The Inline Data Wizard dialog box

The **Inline Data Wizard** dialog box (Figure 59) is opened by clicking the **Inline Wizard...** button in the **Data** page of the **Tool Pane** in the **Edit Script** dialog box (Figure 51 on page 217). It is used to create a **load inline** statement in the script.

The dialog contains something looking like a spreadsheet and in fact works like one in most respects. Note however that calculation formulas will not be evaluated in this spreadsheet as they would e.g. in Microsoft Excel.

Each column represents a field to be loaded into QlikView. Each row is a record in the table to be loaded. You select a data cell by clicking it. A value may then be entered. Press enter or an arrow key to accept the value and move to another cell. Note that empty cells in load inline statements represent 'empty string'. Data can also be pasted into one or more cells from the clipboard.

The dialog contains the following buttons:

OK

The dialog is closed and a **load inline** statement reflecting the contents of the dialog's table grid is generated.

Cancel

Closes the dialog box without generating a load inline statement.

Help

Invokes online help.

A number of commands are available in two menus:

EDIT menu

Cut

Exports the selected cells to the Clipboard. This command is also available on the dialog toolbar. Equivalent to pressing CTRL+X.



Copy

Copies the selected cells to the Clipboard. This command is also available on the dialog toolbar. Equivalent to pressing CTRL+C.



Paste

Pastes the selected cells from the Clipboard. This command is also available on the dialog toolbar. Equivalent to pressing CTRL+V.



Insert Column

Inserts a new blank column if one or more columns are selected.

Delete Column

Deletes a selected column. One or more whole columns must be selected for this command to be available.

Insert Row

Inserts a new blank row if one or more rows are selected.

Delete Row

Deletes a selected row. One or more whole rows must be selected for this command to be available.

TOOLS menu

Document Data

Opens the **Import Document Data Wizard** from which it is possible to paste field values from any existing field in the document. In the dialog you can see how many field values are present in any

selected field. You may also choose to insert all values, selected values or excluded values into the inline wizard.

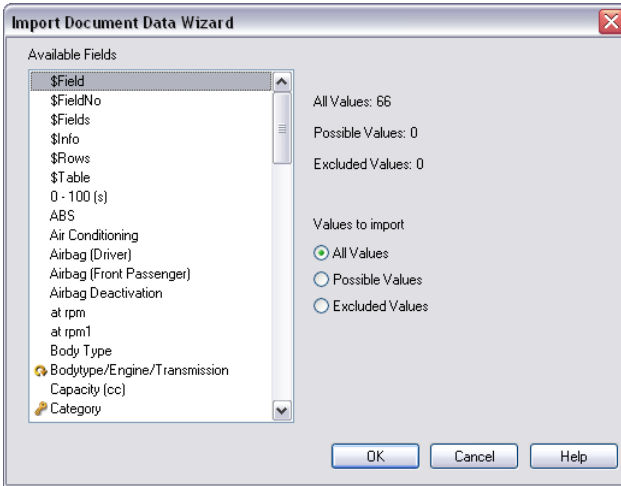


Figure 60. The Import Document Data Wizard

20.10 Access Restriction Table Wizard

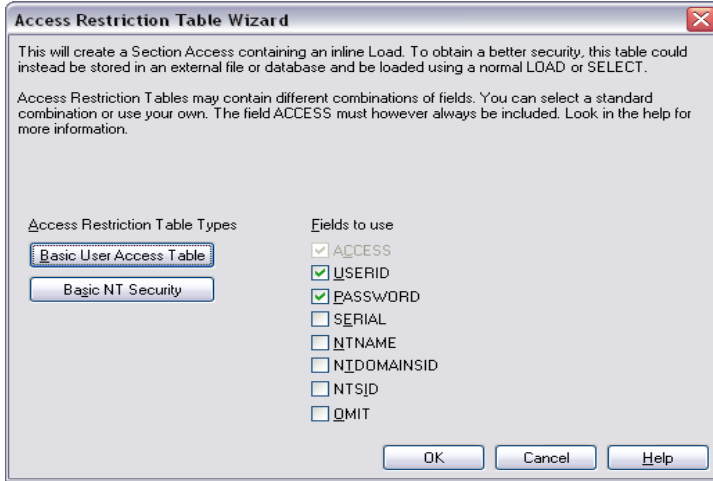


Figure 61. The Access Restriction Table Wizard

The **Access Restriction Data Wizard** dialog box (Figure 61) is opened by clicking the **User Access...** button in the **Data** page of **Tool Pane** in the **Edit Script** dialog box (Figure 51 on page 217). It is used to generate a **load inline** statement for access control to the QlikView document. Read more about security on page 505.

The dialog contains the following commands:

Fields to Use

The list contains all the possible reserved fields in a QlikView **section access**. Mark the check box for those you wish to include.

Basic User Access Table

Press this button to mark ACCESS, USERID and PASSWORD while unmarking the other fields.

Basic NT Security

Press this button to mark ACCESS, PASSWORD, NTNAME and NTDOMAINSID while unmarking the other fields.

OK

The **Inline Data Wizard** dialog (see Figure 59 on page 241) will be opened with the selected fields as column labels. After you press **OK** in this dialog a **load inline** statement preceded by a **section access** statement and succeeded by a **section application** statement will be created.

Cancel

Closes the dialog without generating a load inline statement.

Help

Invokes online help.

20.11 File Wizard

The **File Wizard** automatically appears when you open a file or an HTML table using the **Table Files...** button in the **Data** page of the **Tool Pane** in the **Edit Script** dialog (Figure 51 on page 217). The wizard also opens when a non-qvw file is opened from the **File** menu.

File types that QlikView can recognize are delimited text files (e.g. csv files), fix record files, dif files, Excel files, HTML files and XML files. In this manual these file types are referred to as **Table Files**.

Files can be loaded from the local network or directly from the Internet.

To load a file stored on a local network, use the **Table Files** button in the **Edit Script** dialog. If the file has a file extension of a table file, e.g. .csv, .txt, .tab, .skv, .fix, .dif, .htm, .html, .shtml, .xhtml, .php or .asp, the file will be shown in the **Open Local Files** dialog. It is possible to open the file even if it has a different extension; just set **Files of Type** to **All files (*.*)** in the **Open Local Files**. The content must however still be such that QlikView can interpret it. Whether or not QlikView can, is a different matter from what extension the file has.

To load a file directly from the Internet, paste the URL (previously copied from your web browser) into the text box named **File Name** in the FTP browsing dialog and click **Open**.

Once a file has been selected, the file wizard opens. It contains five pages: **Source**, **Type**, **Transform**, **Options** and **Script**. Since one very rarely needs to change anything on the **Source** page, the wizard starts on the **Type** page. The **<<Back** and the **Next>>** buttons will take you from one page to another.

On the **Type** and **Options** pages there is a preview in which you can see how QlikView has interpreted the file. The **Script** page lets you set how the script is presented in the Edit Script dialog.

The **Finish** button will close the wizard and generate a **load** statement in the load script.

Source Page

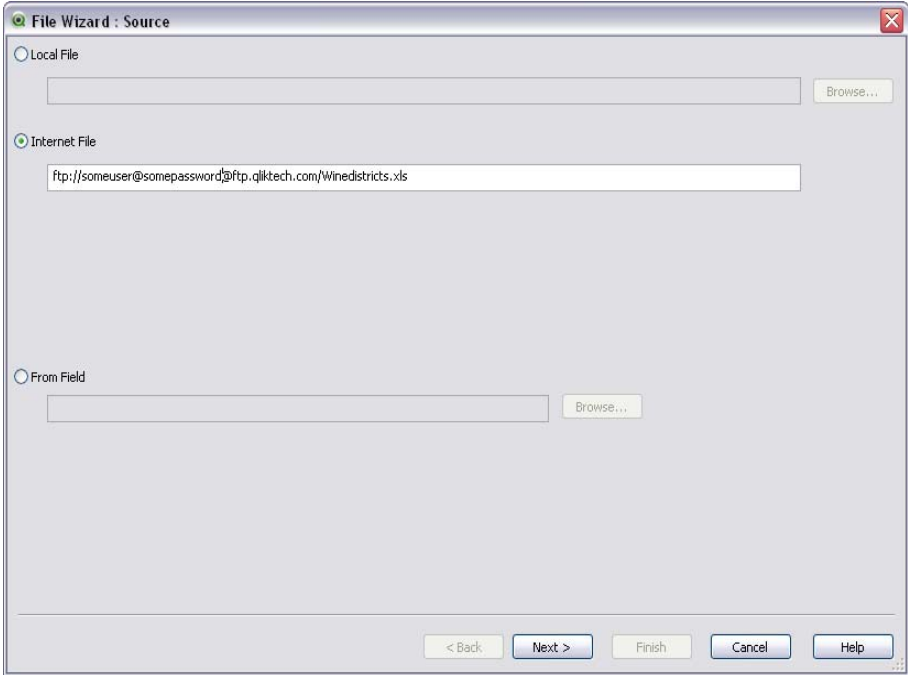


Figure 62. The Source page of the File Wizard

This page contains all the settings related to the source of the table. The source usually is well defined as soon as a file is selected in the **Open File** dialog, but if one needs to make changes in the source definition, which is often the case for e.g. tables fetched directly from Internet pages, this is the place to make them. Pressing the **<<Back** button on the **Type** page opens the **Source** page.

Local File

Enter the path to local file or click **Browse** to find it through an explorer.

Internet File

Enter the path to the Internet file.

From Field

Opens the **Document Data Source** dialog, where you can choose table and field as source. See below for details.

Document Data Source Dialog

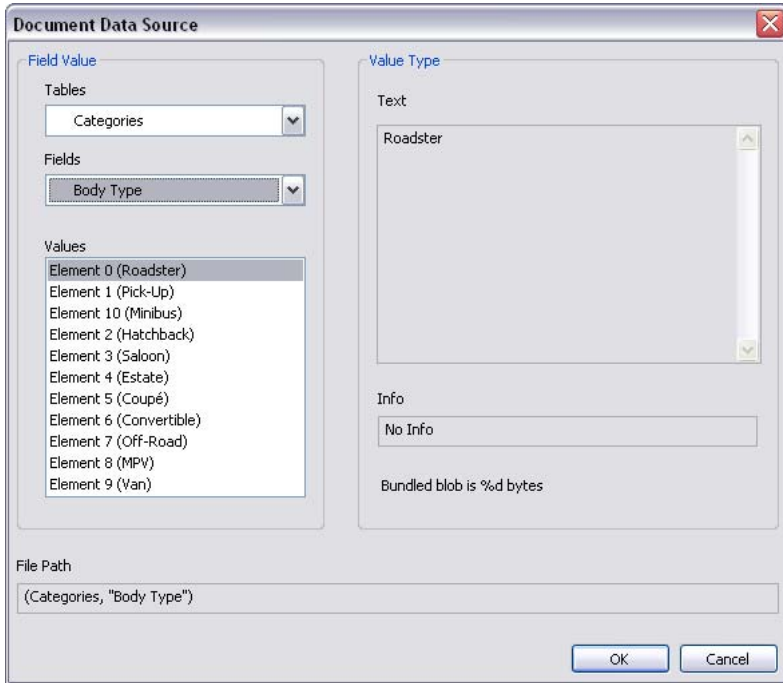


Figure 63. The Document Data Source dialog

This dialog lists the tables of the active document and makes it possible to use a previously loaded field as data source.

Field Value

Tables

A drop-down menu showing the tables that are loaded in the active document.

Fields

The fields of the selected table.

Values

The values of the selected field.

File Path

The name of the selected table and field.

Value Type

The type of the selected value.

Text

If the value is text it is displayed here.

Info

If information is connected to the field value it is displayed here.

Blob

If a blob is connected to the value it is displayed here.

CGI Encoder Dialog

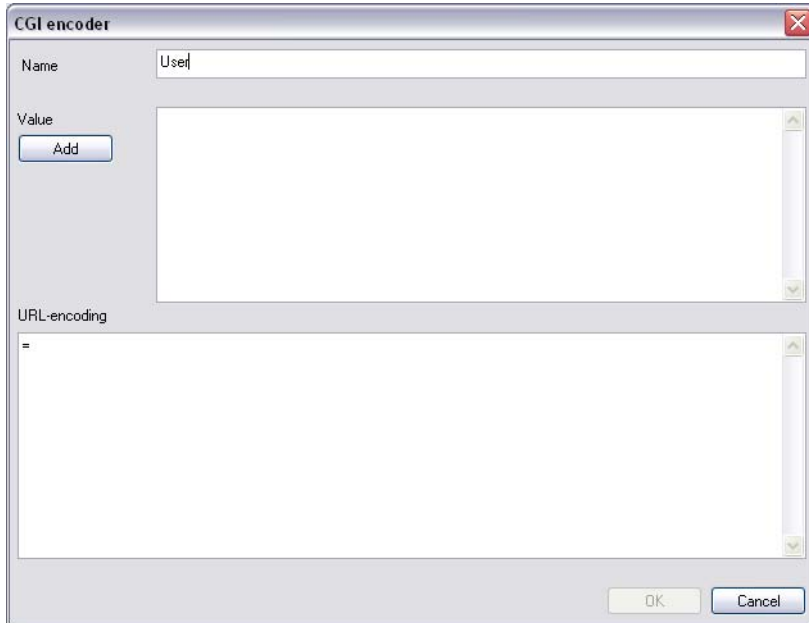


Figure 64. The CGI Encoder dialog

If an HTTP call should be made from a form, this dialog will help you create a correct URL-encoding for the call.

Name

The name of the form field.

Value

The value in the form field

Add

Adds the combination of **Name** and **Value** to the URL encoding list.

URL Encoding

This is the URL-encoding of the field name and field value. This text string should, together with a question mark, be appended to the HTTP call, e.g. “http://www.qlikview.com/data.asp?Field-name=Fieldvalue”.

The **OK** button as well as the **Cancel** button closes the dialog without adding anything to the script.

HTTP Request Dialog

Figure 65. The HTTP Request dialog

An HTTP request is sometimes necessary in the call. This dialog creates some standard requests that easily can be modified for different purposes.

Examples**Group**

In this drop-down it is possible to select different request groups.

Examples

Some typical requests from the selected request group are shown in this drop-down list.

Paste

Pastes the selected example into the edit box. This text can then be edited before the request is added to the request list.

Clear

Clears the content of the edit box.

The **OK** button adds the content of the edit box to the request list and closes the dialog. The **Cancel** button closes the dialog without changing the request list.

Type Page

This page contains the settings related to file type and table type.

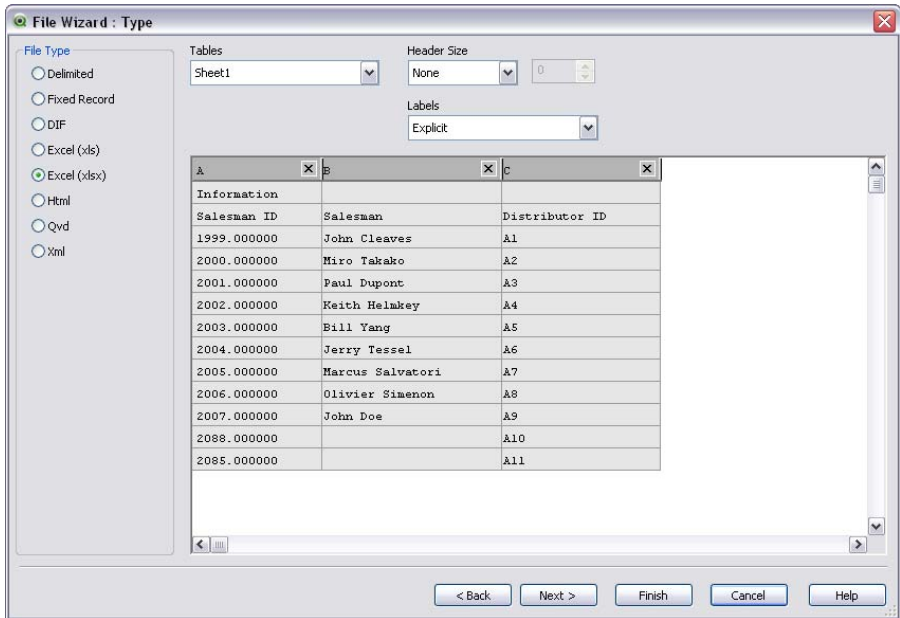


Figure 66. The Type page of the File Wizard

The settings vary according to file type.

When opening a table file, the program tries to interpret the contents of the file. The initial values of the parameters are set this way, but can of course be changed manually by the user. The following groups are the same for all file types:

File Type

You set the type of file in the **File Type** group - **Delimited**, **Fixed record**, **DIF**, **Excel Files (xls)**, **Excel (xlsx)**, **HTML**, **QVD** or **XML**.

Preview Pane

The result of the chosen settings is displayed in the preview pane (the lower part of the dialog page). When the settings are changed, the contents of the preview pane is changed accordingly.

It is also possible to make some changes directly in the preview pane. First, it is possible to omit fields by clicking the “cross” button to the right of the field name. The cross will then change into an exclamation mark and the field will not be included in the **load** statement. To revoke this change, just click the exclamation mark once more.

If you wish to rename a field, just click the field name, then enter the new name in this box. The **as** clauses in the **load** statement required are then created automatically.

Finally, for fix record files (and some HTML tables) it is also possible to set the column widths directly by clicking in the preview pane.

Delimited

The screenshot shows a dialog box titled "Delimited" with the following settings:

- Delimiter:** Comma
- Quoting:** MSQ
- Header Size:** None
- Char Set:** Western European
- Comment:** (empty text box)
- Labels:** None
- Ignore EOF:**

Figure 67. The File Wizard - Type page, delimited file

Delimiter

Here you can specify what type of delimiter is used. If the delimiter is not one of the predefined, choose **Custom** to find the correct delimiter. **Custom** opens the **Delimiter Selector** dialog, where you can choose among several common delimiters.

Header Size

In **Header Size** you can choose to omit the header part of the file: the first line, a specified number of lines or bytes (only for delimited, fixed record and Excel files).

Char Set

The character set of the table file is set here. If the desired character set is not available in the drop-down menu, you can choose **Custom**. **Custom** opens the **Codepage selector** dialog, where you can choose among the character sets installed on the computer.

Quoting

Set the quoting scheme to be used. Choose between **Standard**, **None** and **MSQ**.

Comment

In some files, comments are used between records. Often a comment line begins with a special character or a set of characters, e. g. `//`. Here it is possible to specify the character(s) that mark the beginning of a comment.

Labels

If the field names (column headers) are stored in the first line of the table, this control should be set to **Embedded**. The format also allows field names to be explicitly specified, and if that is the case, this control should be set to **Explicit**. If the table contains no field names, the **None** alternative should be used.

Ignore EOF

Mark this check box if you want the end of file to be ignored.

Fixed Record File

The screenshot shows a configuration window for a fixed record file. It contains several controls:

- Header Size:** A dropdown menu set to 'None' and a numeric spinner set to '0'.
- Record Size:** A dropdown menu set to 'None' and a numeric spinner set to '0'.
- Char Set:** A dropdown menu set to 'Western European'.
- Tab Size:** A numeric spinner set to '1'.
- Buttons:** 'Embedded Labels...', 'Analyze Fix Positions', and 'Clear Fix Positions'.
- Ignore EOF:** An unchecked checkbox.

Figure 68. The File Wizard - Type page, fixed record file

Header Size

In this drop down you can choose to omit the header part of the file: the first line, a specified number of lines or bytes (only for delimited, fixed record and Excel files).

Record Size

Here you can specify the size of a record to be one line, a specified number of lines or a specified number of bytes (only for fixed record files).

Char Set

The character set of the table file is set here. If the desired character set is not available in the drop-down menu, you can choose **Cus-**

tom. Custom opens the **Codepage selector** dialog, where you can choose among the character sets installed on the computer.

Ignore EOF

Mark this check box if you want the end of file to be ignored.

Tab Size

The length of a tab set in spaces.

Embedded Labels...

Embeds the labels. Should be used if the field names are stored in the first line of the table.

Analyze Fix Positions

Analyzes and sets the columns widths for fix record files and for some HTML tables. After the analysis, column breaks can be added or removed in the preview pane.

Clear Fix Positions

Clears all column breaks (for fix record files and for some HTML tables).

DIF

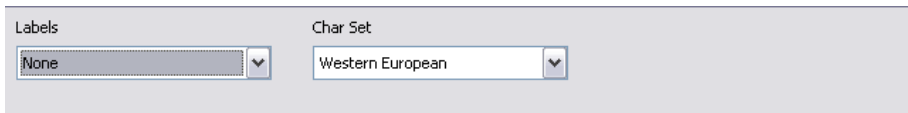


Figure 69. The File Wizard - Type page, DIF file

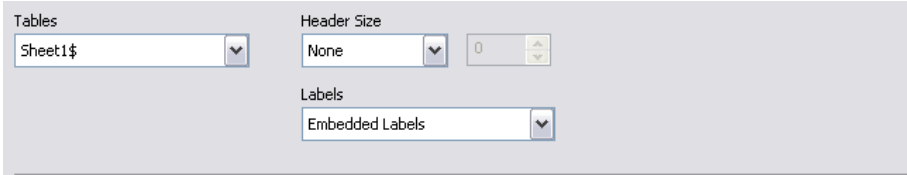
Labels

If the field names (column headers) are stored in the first line of the table, this control should be set to **Embedded**. The **DIF** format also allows field names to be explicitly specified, and if that is the case, this control should be set to **Explicit**. If the table contains no field names, the **None** alternative should be used.

Char Set

The character set of the table file is set here. If the desired character set is not available in the drop-down menu, you can choose **Custom**. **Custom** opens the **Codepage selector** dialog, where you can choose among the character sets installed on the computer.

Excel (xls)



The screenshot shows a dialog box with three main sections. The first section, labeled 'Tables', contains a dropdown menu with 'Sheet1\$' selected. The second section, labeled 'Header Size', contains a dropdown menu with 'None' selected and a numeric input field with '0'. The third section, labeled 'Labels', contains a dropdown menu with 'Embedded Labels' selected.

Figure 70. The File Wizard - Type page, XLS file

Tables

In the **Tables** group you can choose a specific table when reading from a file containing several tables, e.g. HTML or Excel files. In Excel, all sheets in a workbook and all named areas (e.g. multiple cells with names) in the worksheets are identified as possible tables.

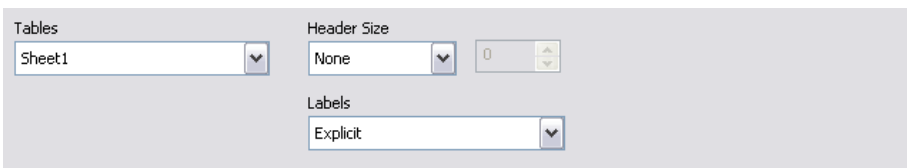
Header Size

In this drop down you can choose to omit the header part of the file: the first line, a specified number of lines or bytes (only for delimited, fixed record and Excel files).

Labels

If the field names (column headers) are stored in the first line of the table, this control should be set to **Embedded**. The format also allows field names to be explicitly specified, and if that is the case, this control should be set to **Explicit**. If the table contains no field names, the **None** alternative should be used.

Excel (xlsx)



The screenshot shows a dialog box with three main sections. The first section, labeled 'Tables', contains a dropdown menu with 'Sheet1' selected. The second section, labeled 'Header Size', contains a dropdown menu with 'None' selected and a numeric input field with '0'. The third section, labeled 'Labels', contains a dropdown menu with 'Explicit' selected.

Figure 71. The File Wizard - Type page, XLSX file

Tables

In the **Table** group you can choose a specific table when reading from a file containing several tables, e.g. HTML or Excel files. In Excel, all sheets in a workbook and all named areas (e.g. multiple cells with names) in the worksheets are identified as possible tables.

Header Size

In this drop down you can choose to omit the header part of the file: the first line, a specified number of lines or bytes (only for delimited, fixed record and Excel files).

Labels

If the field names (column headers) are stored in the first line of the table, this control should be set to **Embedded**. The format also allows field names to be explicitly specified, and if that is the case, this control should be set to **Explicit**. If the table contains no field names, the **None** alternative should be used.

HTML

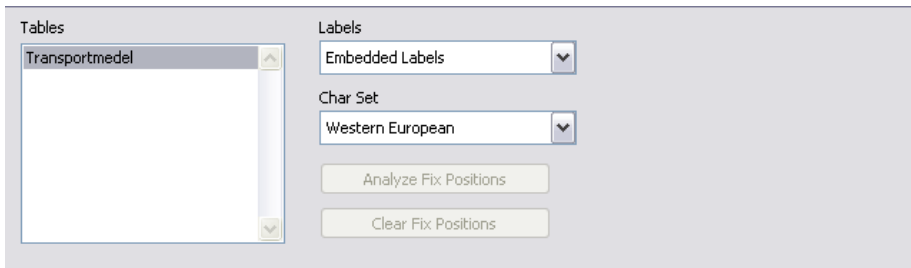


Figure 72. The File Wizard - Type page, HTML file

Tables

In the **Table** group you can choose a specific table when reading from a file containing several tables, e.g. HTML or Excel files. In Excel, all sheets in a workbook and all named areas (e.g. multiple cells with names) in the worksheets are identified as possible tables.

Labels

If the field names (column headers) are stored in the first line of the table, this control should be set to **Embedded**. The format also allows field names to be explicitly specified, and if that is the case, this control should be set to **Explicit**. If the table contains no field names, the **None** alternative should be used.

Char Set

The character set of the table file is set here. If the desired character set is not available in the drop-down menu, you can choose **Custom**. **Custom** opens the **Codepage selector** dialog, where you can choose among the character sets installed on the computer.

Analyze Fix Positions

Analyzes and sets the columns widths for fix record files and for some HTML tables. After the analysis, column breaks can be added or removed in the preview pane.

Clear Fix Positions

Clears all column breaks (for fix record files and for some HTML tables).

QVD

This file type has no configurable settings.

XML

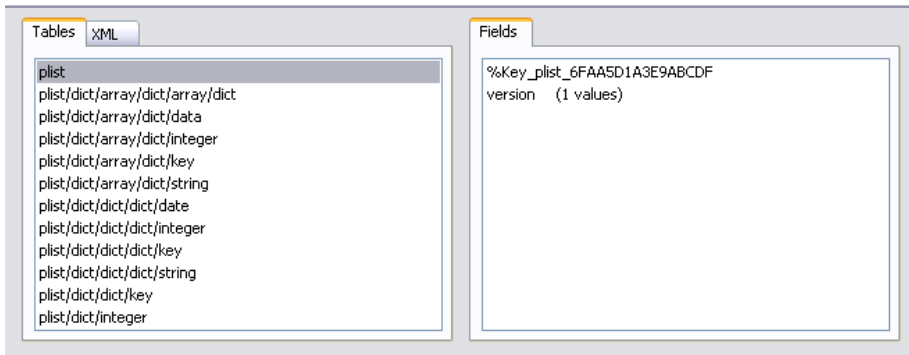


Figure 73. The File Wizard - Type page, XML file

Tables

The found tables are shown in this list. Each table is shown as a path that corresponds where in the XML structure it was found. When a table is selected, its fields will be shown in the **Fields** pane to the right. A load statement will be generated for each table in the list when the **Finish** button is clicked.

XML

This sheet shows the interpreted XML structure.

Fields

This sheet shows the fields of the selected table and the generated keys.

Transform Dialog

To reach the transform dialogs you must enable this step. The step is not enabled automatically in order to save memory.



Figure 74. The Enable Transform page of the File Wizard

In the **Transform** dialog it is possible to filter and make advanced transformations of a table. For HTML files this is important, since there is no commonly used standard for how to store data in HTML. As a result, it is impossible to make QlikView capable of reading HTML tables as easily as other more structured data formats.

The Transform dialog provides a variety of filters that can be used to make tables more suitable for QlikView. The dialog has six pages; **Garbage**, **Fill**, **Column**, **Context**, **Unwrap** and **Rotate**. These are described below. The five button commands to the left in the **Transform Table** dialog are however used for all six pages:

Undo

Undoes the last added change.

Redo

Redoes the last Undo.

Reset

Resets the table to the original state.

OK

Accepts all transformations and closes the dialog.

Cancel

Resets the table to the original state and closes the dialog.

Garbage

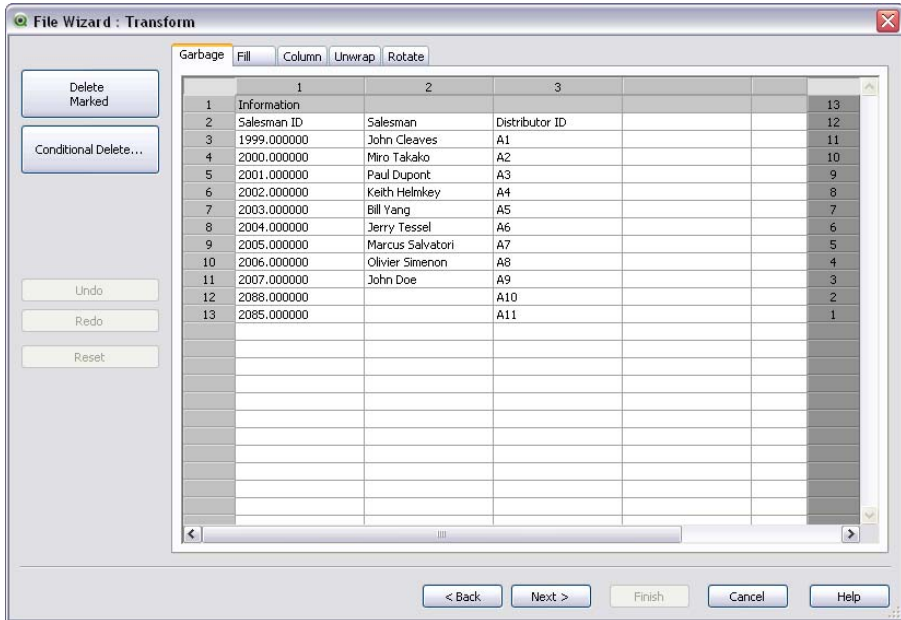


Figure 75. The Garbage tab of the File Wizard : Transform page

On this page, rows and columns containing garbage can be removed from the table. For HTML files it is not uncommon with extra rows and columns that are there just for legibility and these must of course be removed before loading the data into QlikView.

Columns and rows can both be removed explicitly in which case their indices are stored in the script. Rows can also be removed using conditional criteria.

For instance, in the table shown in Figure 75 one would probably want to remove several rows. It is then easy to remove the lines 1, 12 and 13 by marking them and clicking the **Delete Marked** button. Once this has been done, row 2 can be used as labels for the columns.

Delete Marked

Removes marked rows and columns.

Conditional Delete

Opens the **Specify Row Condition** dialog where the conditional criteria for deletion of rows can be set.

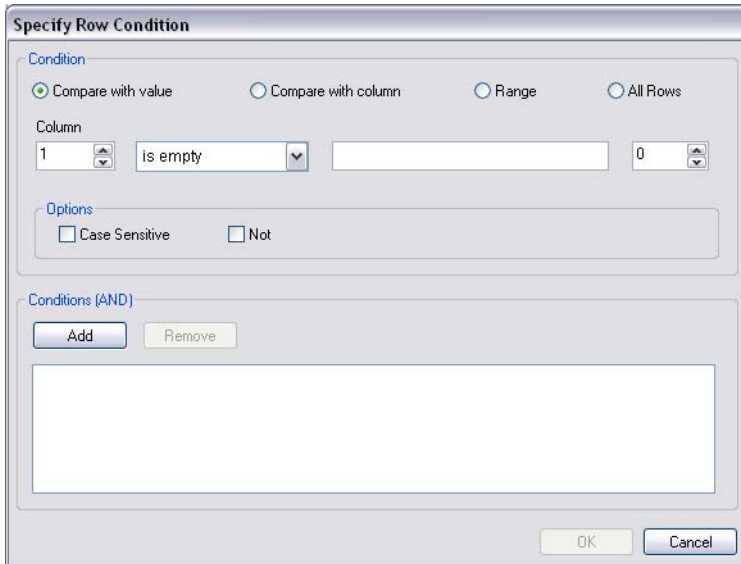
Specify Row Condition

Figure 76. The Specify Row Condition dialog

This dialog can be opened via the **Garbage** page (the button **Conditional Delete**), the **Column** page (press **New..** and then **Cells from these rows**) or the **Unwrap** page (the button **Conditional Unwrap...**). Here it is possible to define a logical condition for selection of one or several rows. Depending on the context the rows that are defined are either deleted, copied to a new column and split the table in several parts.

Condition

In the **Condition** group you can specify a condition where a column equals a specific value or another column, a range of records or all records. Several conditions can be applied together. After defining a condition you must press the **Add** button to include it in the transformation.

Compare with Value

Matches the cell with a cell condition that can evaluate to true or false.

Compare with Column

Matches the cell with the corresponding cell in another column.

Range

Selects x rows, then skips y rows. Starting and ending positions for this may be specified with indexes.

All Rows

Use this condition if all rows should be deleted.

Column

Sets the column number on which the condition should be applied.

From

Only visible in **Range** mode. Sets the row number from which rows should be deleted or a second condition should be applied.

To

Only visible in **Range** mode. Sets the row number to which rows should be deleted or a second condition should be applied.

Select

Only visible in **Range** mode.

Here it is possible to set whether rows should be selected cyclically or skipped cyclically, e.g. cyclically select two rows, then skip one. If all rows should be used, Select should be set to 1 and Skip should be set to 0.

Options

In the **Options** subgroup you can specify modifiers for the selection.

Case Sensitive

Mark this check box if comparisons should be case sensitive.

Not

Mark this check box if the selection criteria should be inverted.

Conditions (AND)

In the **Conditions (AND)** group you see the conditions specified and included in the transformation. A logical AND is applied between conditions.

Add

Adds the current condition to the list.

Remove

Removes a selected condition from the list.

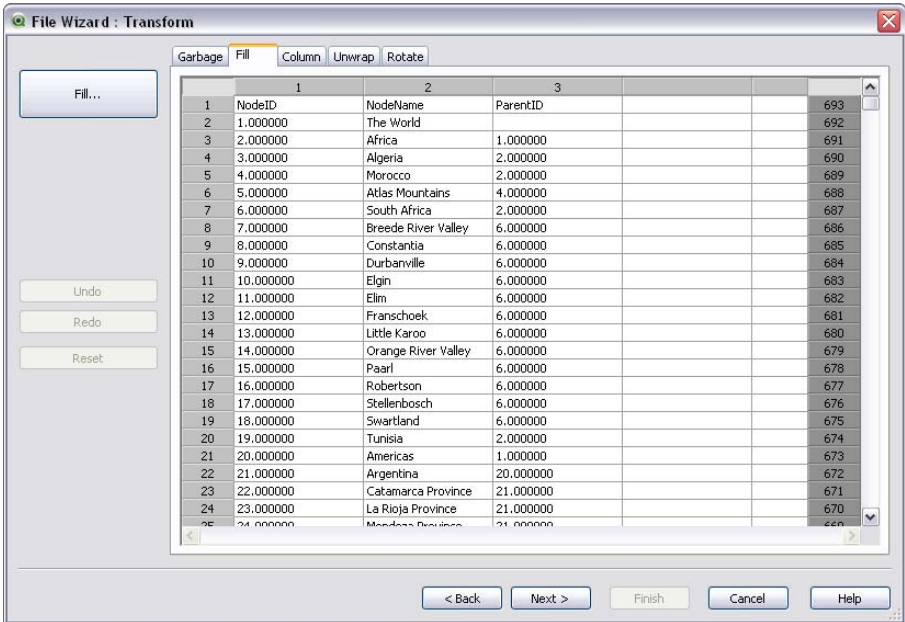
Fill

Figure 77. The Fill tab of the File Wizard : Transform page

With this page you can replace cell values. The power of this feature will become evident as you use the wizard. In most cases certain cells that match a specific description are replaced with another value from the same column.

A common case is that cells are empty and that the first non-empty cell above contains the relevant value.

Fill Cells Dialog

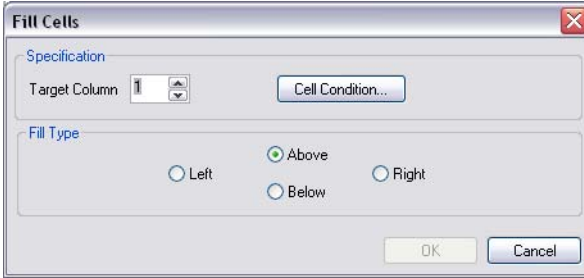


Figure 78. The Fill Cells dialog

Specification

In the **Specification** group it is possible to specify in what cases the cell should be filled.

Target Column

Sets the column number on which the condition should be applied.

Cell Condition

Opens the cell condition dialog where it is possible to set the cell condition.

Fill Type

Sets the strategy for how cells should be filled. Use one of the values **Above**, **Left**, **Right** or **Below**.

Cell Condition Dialog



Figure 79. The Cell Condition dialog

This dialog opens from the **Fill Cells** dialog and offers a possibility to specify a logical condition that refers to the contents of a cell.

Cell Value

There are several logical options in the drop-down menu. To compare strings, enter a string to be compared in the field after. To compare the length of the cell contents, enter the comparison length (a numeric value) to the far right.

Not

Activate this option if the condition should be reversed using a logical NOT.

Case Sensitive

Activate this option if you want the value comparison to be case sensitive.

Column

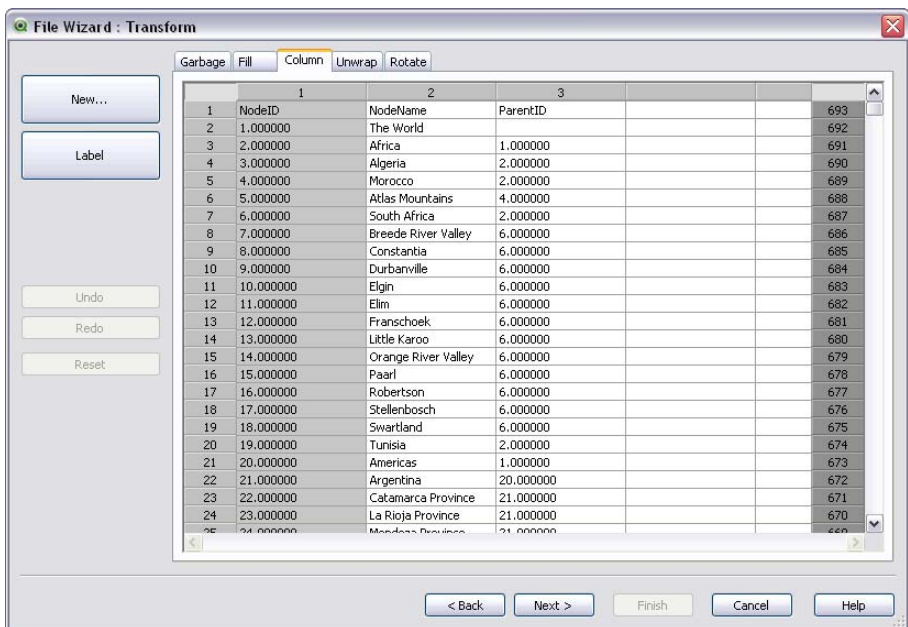


Figure 80. The Column tab of the File Wizard : Transform page

This page is used to copy a column to a new column. The new column may contain all the cells of the source column or a selection of the cells.

This feature can be useful if a column contains different kinds of values. It can also be used to duplicate a column that contains HTML links, so that one column contains the link's text, whereas the other column contains its URL.

New

Opens the **Specify Cells for New Column** dialog where it is possible to specify how the new column should be created.

Label

Sets the label for the column.

Specify Cells for New Column

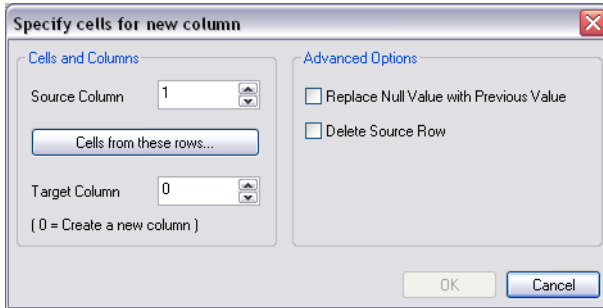


Figure 81. The Specify Cells for New Column

Cells and Columns**Source Column**

Sets the column number from which cell values should be copied.

Cells from these rows

Opens the **Specify Row Condition** dialog (see page 259) where it is possible to set the Row condition.

Target Column

Specifies the number of the target columns.

Advanced Options**Replace Null Value with Previous Value**

Fills empty cells with the value in the cell above.

Delete Source Row

Deletes the source row.

Context

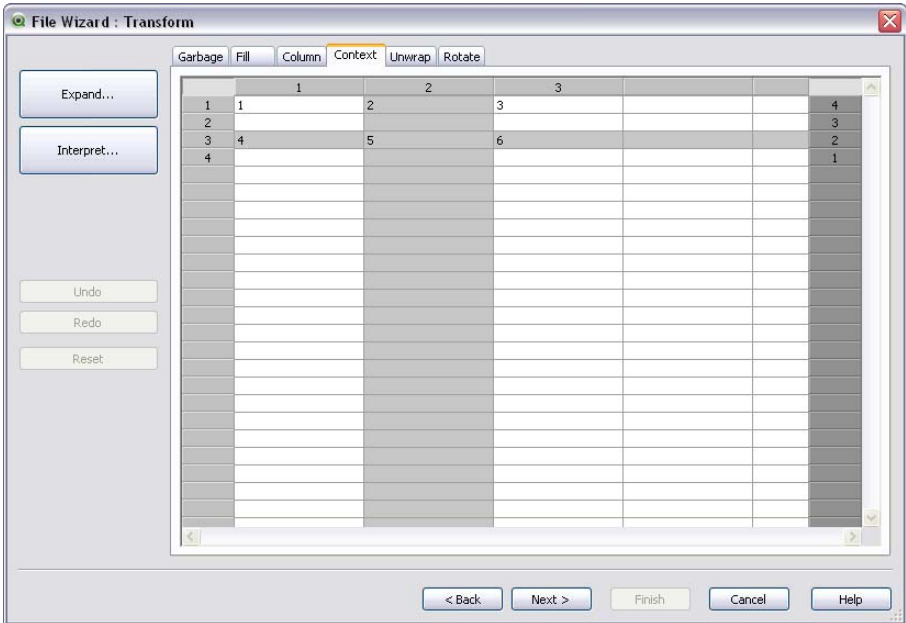


Figure 82. The Context tab of the File Wizard : Transform page

In an html table more information than what is visible can be stored, e.g. in the case that the content of a cell is clickable, the address to where the browser should jump must also be stored. The **Transform** page will show the text of the cell, but is also able to show the additional information belonging to the different cells. Naturally, it is also possible to read this information into QlikView.

Tags always enclose the additional information. A tag has a name and may have attributes, which sometimes have values. The context of a cell may look like this:

Example:

```
<A href=www.myurl.com/mypage.html name="MyName">
My link text
</A>
```

The visible text of the cell is *My link text*. The context contains two attributes, a start tag and an end tag. The tags specify a hypertext link. The start tag has two attributes, *href* and *name*.

The **Context** page can be used to read the attribute of tags instead of the main string. Other kinds of operations are also possible. You can right-click the cells in the wizard and choose **View context** to look at the context of the cell.

Expand

Opens the **Context cell expansion** dialog in which an expansion of the contents of one cell into several cells can be defined. Note that both a column and a row must be marked in order for this button to be enabled. However, all the cells in the column will be expanded.

Interpretation

Opens the **Context Interpretation** dialog in which an interpretation of the contents of the cells can be defined. Note that both a column and a row must be marked in order for this button to be enabled. However, all the cells in the column will be interpreted.

Context Cell Expansion

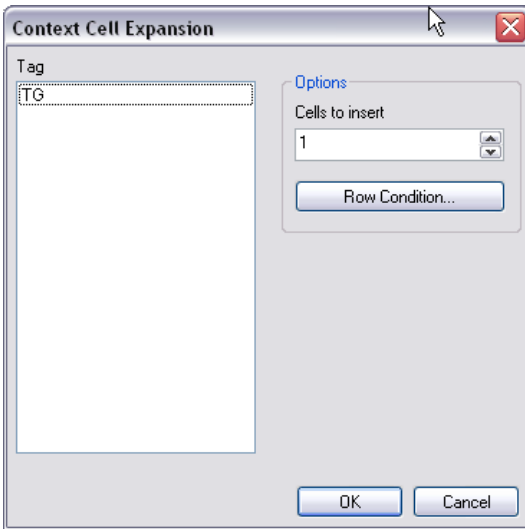


Figure 83. The Context Cell Expansion dialog of the File Wizard

With this dialog it is possible to expand the contents of one cell into several cells. In some situations you will find that several rows in a column is stored within one cell of the HTML table, i.e. between the `<TD>` and the `</TD>` tags. This cell can then be expanded into a column. Just mark the tag that is used as delimiter between the different rows within the column. Usually this is the `
` symbol.

Cells to Insert

Number of cells to insert. Normally this should be set to 1, but if extra rows are needed, this control should be set to a higher number.

Row Condition

Condition for what rows to expand.

Context Interpretation

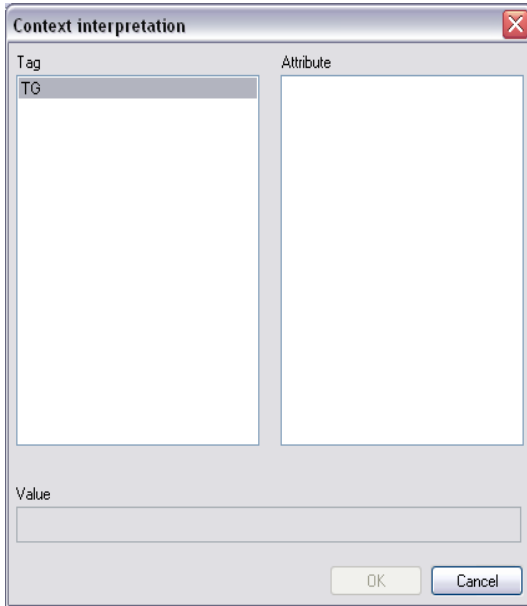


Figure 84. The Context Interpretation dialog of the File Wizard

With this dialog it is possible to exchange the value of a cell with piece of hidden information found in the cell.

The cell can contain several other tags, each with one, sometimes several, attributes. By selecting the appropriate tag and attribute and then pressing OK, the cell content is replaced with that of the value of the attribute.

Tag

A list of the tags found in the cell.

Attribute

The attribute of the marked tag.

Value

The value of the attribute.

Unwrap

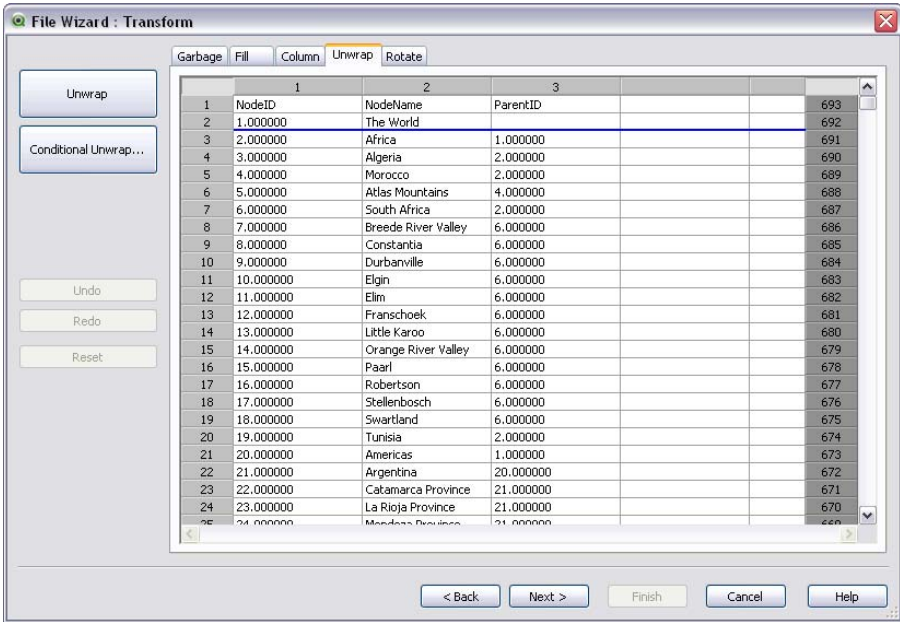


Figure 85. The Unwrap tab of the File Wizard : Transform page

Here you can straighten out a wrapped table. “Wrapped” here means that a second part of the table continues, not below as it should, but beside the first part of the table. In the example above, the right half will thus be moved and put below the left half.

Unwrap

Unwraps the table. The border between the two halves must be set with the cursor first. It is possible to split the table either vertically or horizontally.

Conditional Unwrap

Defines the condition for a vertical split of the table. Opens the **Specify Row Condition** dialog, see page 259.

Rotate

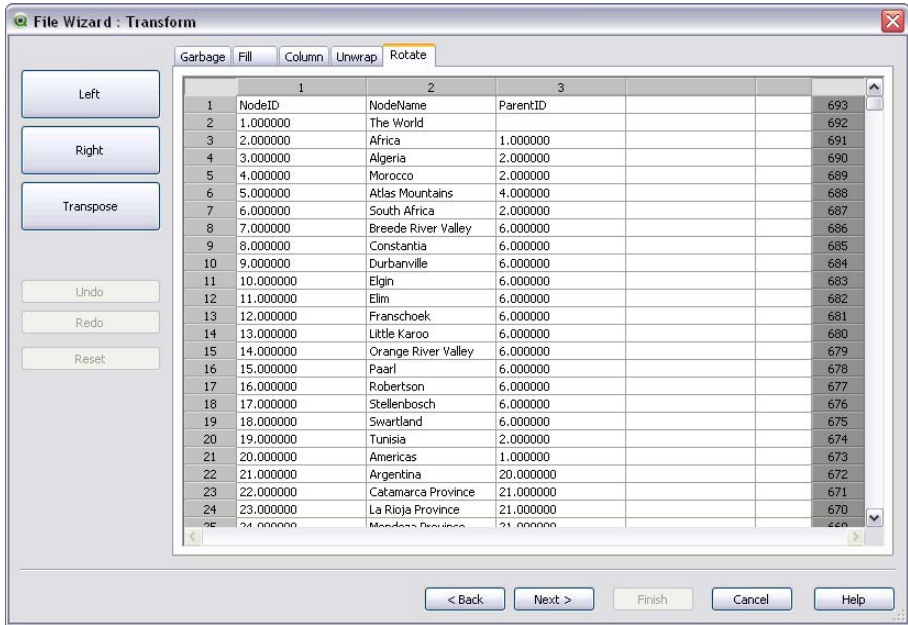


Figure 86. The Rotate tab of the File Wizard : Transform page

Web designers often rotate tables to give them a nicer look. The main intention of the rotation feature is to make it possible to turn tables "back to normal"; to get fields as columns, field names on the first row etc.

Left

Rotates the table counterclockwise.

Right

Rotates the table clockwise.

Transpose

Transposes the table, i.e. turns the table around so that the top right cell becomes the bottom left cell and the bottom left becomes the top right. The top left and bottom right cells, however, stay in place.

Options

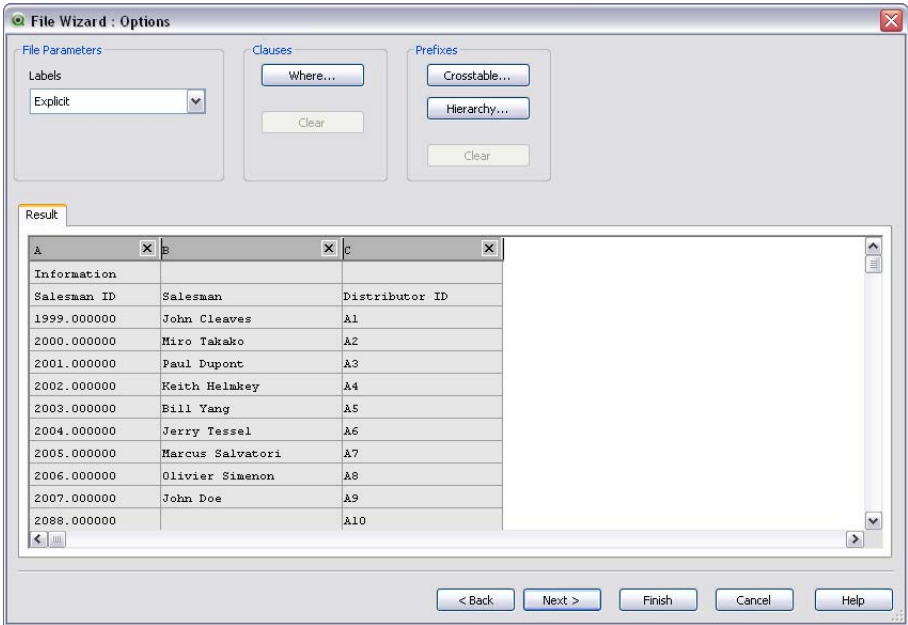


Figure 87. The Options page of the File wizard

File Parameters

Labels

If the field names (column headers) are stored in the first line of the table, this control should be set to **Embedded**. The **DIF** format also allows field names to be explicitly specified, and if that is the case, this control should be set to **Explicit**. If the table contains no field names, the **None** alternative should be used.

Clauses

Where...

Opens the **Where Clause** dialog (see below), with which is possible to create a **where** clause for the **load** statement.

Clear

Revokes the transformation and/or the **Where** clause of a cross table.

Prefixes

Crosstable...

Opens the **Cross Table** dialog (see below), where it is possible to set the parameters for a transformation of a cross table into a three- (or more) column table

Hierarchy...

Opens the **Hierarchy Parameters** dialog (see below), where it is possible to set the parameters for a hierarchy table.

Clear

Revokes the transformation of a cross table or a hierarchy table.

Note that if both a cross table transformation and a **where** clause are used, the **where** clause is evaluated first. It is thus not possible to first transform from a cross table and then apply a **where** clause. However, if there is a need to apply a **where** clause on an already transformed table, a construction with a temporary table will solve the problem:

```
TempTable: Crosstable (...) Load ... from ...;
RealTable: Load ... resident TempTable where ...;
Drop Table TempTable;
```

Where Clause

	Field	Op/Func	Constant
<input checked="" type="radio"/> Simple	A	<	
<input type="radio"/> Advanced	A	<	
<input type="radio"/> Empty Template	A	<	

WHERE()

Where(1=1)

OK Cancel Help

Figure 88. The Where Clause wizard

Simple

Choose what **Field(s)** should be part of the where clause and what operator/function should be used. You may also enter a **Constant** based on existing field values. The second and third rows become active when you select **AND** or **OR** in the drop-down list to the left of each row.

Advanced

Mark Advanced if you want to type the clause from scratch.

Empty Template

This option will insert `where (1=1)` into the load script and you can edit it further there.

Crosstable

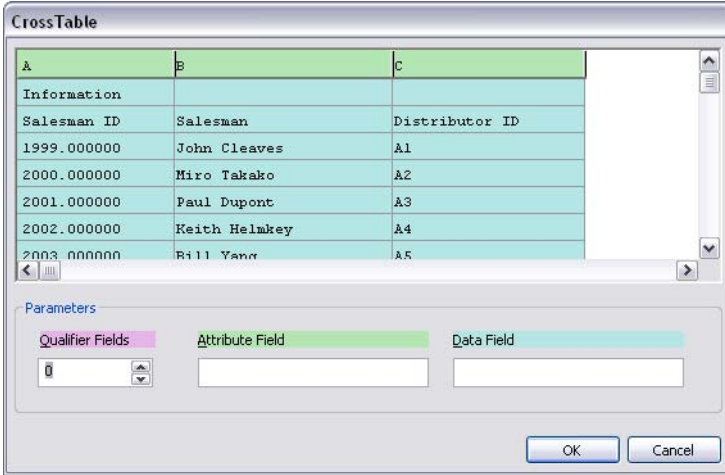


Figure 89. The Crosstable wizard

Parameters

Qualifier Fields

The number of fields that precede the fields to be transformed.

Attribute Name

The name of the new field that will contain all the fields (attribute values) to be transformed.

Data Name

The name of the field that will contain the data of the fields of the attribute field.

Read more about cross tables on page 291 and on page 476.

Hierarchy Parameters

Parameters as they appear in the script are in parentheses.

The screenshot shows a dialog box titled "Hierarchy Parameters". It contains three main sections:

- Source Parameters:** Three dropdown menus labeled "ID Field", "Parent ID Field", and "Name Field".
- Hierarchy Parameters:** A checked checkbox followed by five text input fields: "Parent Name", "Path Name", "Depth Name", "Path Source", and "Path Delimiter".
- Belongs To Parameters:** A checked checkbox followed by three text input fields: "Ancestor Id", "Ancestor Name", and "Depth Diff Name".

At the bottom right, there are "OK" and "Cancel" buttons.

Figure 90. The Hierarchy parameters dialog

Source Parameters

ID Field (*NodeID*)

The name of the field that contains the node id.

Parent ID Field (*ParentID*)

The name of the field that contains the node id of the parent node.

Name Field (*NodeName*)

The name of the field that contains the name of the node.

Hierarchy Parameters

Parent Name (*ParentName*)

The name of the new **ParentName** field. Optional parameter.

Path Name (*PathName*)

The name of the new **Path** field, which contains the path from the root to the node. Optional parameter.

Depth Name (*Depth*)

The string used to name the new **Depth** field, which contains the depth of the node in the hierarchy. Optional parameter.

Path Source (*PathSource*)

The name of the field that contains the name of the node used to build the node path. Optional parameter. If omitted **NodeName** will be used.

Path Delimiter (*PathDelimiter*)

The string used as delimiter in the new **Path** field, e.g. \. Optional parameter. If omitted '/' will be used.

Belongs To Parameters**Ancestor ID** (*AncestorID*)

The name of the new **ancestor id** field, which contains the id of the ancestor node.

Ancestor Name (*AncestorName*)

The name of the new **ancestor** field, which contains the name of the ancestor node.

Depth Diff Name (*DepthDiff*)

The name of the new **DepthDiff** field, which contains the depth of the node in the hierarchy relative the ancestor node. Optional parameter.

Read more about hierarchies on page 299 and page 481.

Script

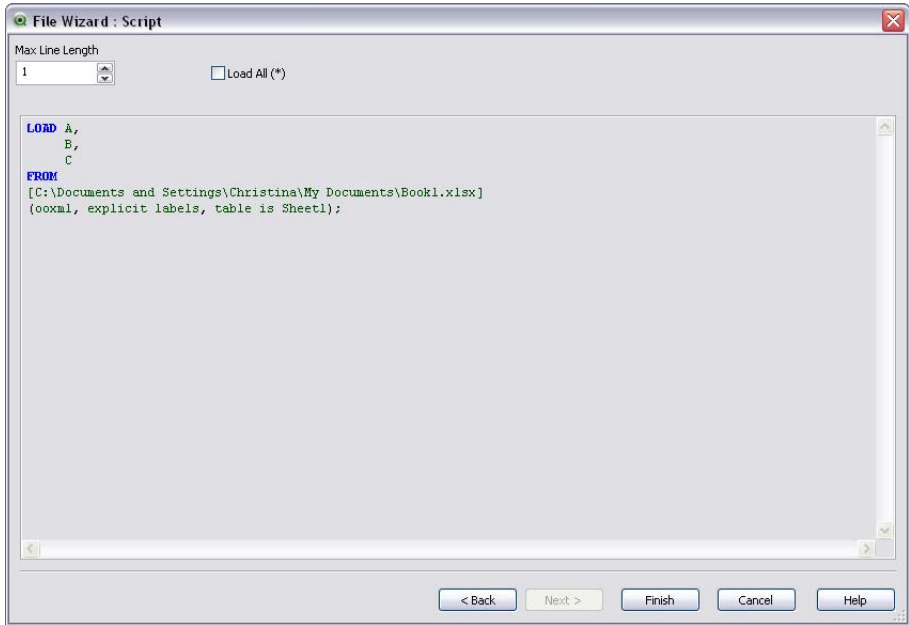


Figure 91. The Script page of the File Wizard

On the **Script** page you can set **Max Line Length** to make the script easier to interpret. Mark the check box **Load All** to load all fields in the table.

The **Finish** button, finally, generates the **load** statement and closes the wizard.

Debugger

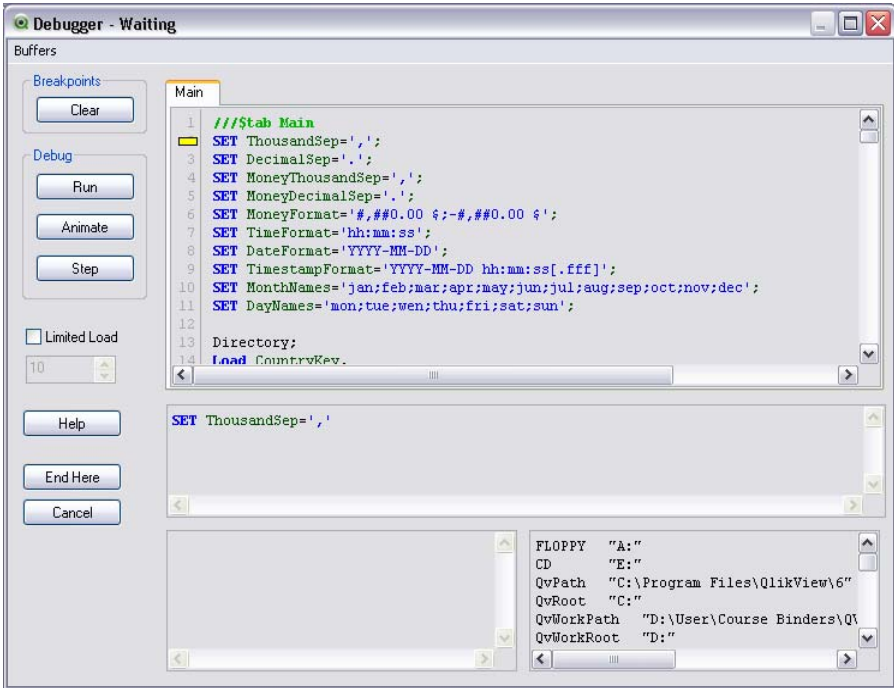


Figure 92. The Script Debugger

With the script debugger it is possible to search for errors in the script. Every script statement can be monitored and the values of variables can be examined while the script is executed.

The top window shows the script. The position of the execution is marked by a yellow cursor. Breakpoints can be set by clicking the number to the left of a statement. The breakpoint is marked by a red dot.

The middle window shows the statement that is about to be executed (step mode) or being executed (run mode).

The left bottom window shows error and status codes, essentially the same information as in the script progress dialog.

The right bottom window shows all the variables and their respective values. Changed variables are shown in red.

Breakpoints

In the **Breakpoints** group breakpoints that have been set in the script window to the right can be cleared. A breakpoint is shown as a red circle.

Clear

Clears all the breakpoints.

Debug

The **Debug** group contains the buttons that determine how the script is run in the Debugger.

Run

Runs the script. The script execution will halt when a breakpoint is encountered.

Animate

Runs the script, but with a pause after each statement. This way it is possible to get an overview of the script execution, without interacting with the mouse.

Step

Executes one statement of the script.

Limited Load

Limits the **load** sequences to the first N accepted records at each **load** and **select** statement. N is a number that is set in the box below.

Note This setting has no effect on a BINARY load. All data will be loaded even if this check box is marked.

Help

Opens the QlikView help program.

End Here

Closes the Debugger, but lets QlikView keep the data loaded so far.

Cancel

Aborts the script execution and discards all data loaded.

Buffers

In the **Buffers** menu you may open tabs showing external script files referenced via **\$(include)**.

Open

Displays a dialog where you may open a script file. The contents of the file will be displayed under a separate tab and can be used for debugging.

Close

Closes the current debugger script tab. The Main tab cannot be closed.



21 SCRIPT SYNTAX

In the script, the name of the data source, the names of the tables and the names of the fields included in the logic are defined. Furthermore, the fields in the access rights definition are defined in the script.

The script consists of a number of statements that are executed consecutively.

For a formal description of the BNF syntax notation used in this and the following chapters, see page 523 in Book III.

21.1 Statements and Keywords

The QlikView script consists of a number of statements. A statement can be either a regular statement or a control statement. Certain statements can be preceded by prefixes.

Regular statements are typically used to manipulate data in one way or another. These statements may be written over any number of lines in the script and must always be terminated by a semicolon, ";".

Control statements are typically used for controlling the flow of the script execution. Each clause of a control statement must be kept inside one script line and may be terminated by a semicolon or the end-of-line.

Prefixes may be applied to applicable regular statements but never to control statements. The **when** and **unless** prefixes can however be used as suffixes to a few specific control statement clauses.

In the next subchapter you will find an alphabetical listing of all script statements, control statements and prefixes.

All script keywords can be typed with any combination of lower case and upper case characters. Field and variable names used in the statements are however case sensitive.

21.2 Script Keywords and Statements in Alphabetical Order

Add	page 284
Alias.....	page 285
Binary	page 285

Buffer.....	page 286
Bundle	page 287
Call	page 288
Concatenate	page 289
Connect.....	page 290
Crosstable	page 291
Directory.....	page 291
Disconnect.....	page 292
Do..loop	page 292
Drop Field.....	page 293
Drop table	page 293
Execute	page 294
Exit script	page 294
First.....	page 295
For..next.....	page 295
For each..next	page 296
Force	page 298
Generic	page 299
Hierarchy	page 299
HierarchyBelongsTo.....	page 300
If..then..elseif..else..end if	page 301
Image_size.....	page 302
Info	page 302
Inner.....	page 303
Inputfield	page 304
Intervalmatch.....	page 304
Intervalmatch (extended).....	page 305
Join	page 306
Keep.....	page 306
Left	page 307
Let.....	page 308

Load.....	page 309
Loosen table	page 318
Mapping.....	page 318
Map..using	page 319
Noconcatenate	page 321
NullAsValue	page 321
NullAsNull	page 322
Outer	page 323
Qualify.....	page 323
Rem	page 325
Rename field.....	page 325
Rename table	page 326
Replace	page 326
Right	page 328
Sample	page 328
Section	page 329
Select	page 329
Semantic	page 332
Set.....	page 332
Sleep	page 333
SQL	page 333
SQLColumns.....	page 333
SQLTables	page 334
SQLTypes	page 335
Star.....	page 336
Store.....	page 336
Sub..end sub	page 337
Switch..case..default..end switch.....	page 338
Trace	page 339
Unless	page 339
Unmap	page 340

Unqualify.....	page 341
When	page 342

Add

The **add** prefix can be added to any **load**, **select** or **map..using** statement in the script. It only has a meaning during partial reloads. During a partial reload, the QlikView table whose table name is generated by the **add load** / **add select** statement (if such a table exists) will be appended with the result of the **add load** / **add select** statement. No check for duplicates is performed. Therefore, a statement using the **add** prefix will normally include either a **distinct** qualifier or a **where** clause guarding duplicates. The **add map..using** statement causes the mapping to take place also during partial script execution.

The syntax is:

```
add [ only ] ( loadstatement | selectstatement |  
map..usingstatement )
```

where

only is an optional qualifier, which denotes that the statement should be disregarded during normal (non-partial) reloads.

Examples:

```
Tab1:  
load Name, Number from Persons.csv;  
add load Name, Number from NewPersons.csv;
```

During normal reload, data is loaded from *Persons.csv* and stored in the QlikView table *Tab1*. Data from *NewPersons.csv* is then concatenated to the same QlikView table.

During partial reload, data is loaded from *NewPersons.csv* and appended to the QlikView table *Tab1*. No check for duplicates is made.

```
Tab1:  
select Name, Number from Persons.csv;  
add load Name, Number from NewPersons.csv  
where not exists(Name);
```

A check for duplicates is made by means of looking if *Name* exists in the previously loaded table data (see the function **exists** under “Inter-Record Functions” on page 412).

During normal reload, data is loaded from *Persons.csv* and stored in the QlikView table *Tab1*. Data from *NewPersons.csv* is then concatenated to the same QlikView table.

During partial reload, data is loaded from *NewPersons.csv*, which is appended to the QlikView table *Tab1*.

A check for duplicates is made by means of looking if *Name* exists in the previously loaded table data (see the function **exists** under “Inter-Record Functions” on page 412).

```
Tab1:
load Name, Number from Persons.csv;
add only load Name, Number from NewPersons.csv
where not exists(Name);
```

During normal reload, data is loaded from *Persons.csv* and stored in the QlikView table *Tab1*. The statement loading *NewPersons.csv* is disregarded.

During partial reload, data is loaded from *NewPersons.csv*, which is appended to the QlikView table *Tab1*. A check for duplicates is made by means of looking if *Name* exists in the previously loaded table data (see the function **exists** under “Inter-Record Functions” on page 412).

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

See also the **replace** keyword on page 326, which is used to replace tables during partial reload.

Alias

The **alias** statement is used to set an alias according to which a field will be renamed whenever it occurs in the script that follows. The syntax is:

```
alias fieldname as aliasname { , fieldname as aliasname }
```

where

fieldname and *aliasname* are strings representing what the respective names imply.

Examples:

```
Alias ID_N as NameID;
Alias A as Name, B as Number, C as Date;
```

The name changes defined through this statement are used on all subsequent **select** and **load** statements. A new **alias** for a field name can be defined by a new **alias** statement at any subsequent position in the script.

Binary

The **binary** statement is used to load the access and data parts of a QlikView document. It does not load the layout information. Only one **binary** statement is allowed in the script and it can only be put as the first statement of a script. The syntax is:

binary file

where

file ::= [path] filename

Examples:

```
Binary customer.qvw;  
Binary c:\qv\customer.qvw;
```

The path is the path to the file, either absolute, or relative to the qvw file containing this script line.

Buffer

QVD files can be created and maintained automatically via the **buffer** prefix. This prefix can be used on most **load** and **select** statements in script. It indicates that a QVD file is used to cache/buffer the result of the statement.

Numerous limitations exist, the most notable is that there must be either a file **load** or a **select** statement in “the foundation” of any complex statement.

The name of the QVD file is a *calculated* name (a 160-bit hexadecimal hash of the entire following **load** or **select** statement and other discriminating info) and is typically stored in the Application Data folder or another folder specified under **User Preferences - Locations** page (see page 120). This means that the QVD buffer will be rendered invalid by any change in the following **load** or **select** statement.

QVD buffers will normally be removed when no longer referenced throughout a complete script execution in the document that created it and when the document that created it no longer exists. This housekeeping can be turned off via settings in the **General** pages of **Document Properties** and **User Preferences** respectively.

The syntax is:

```
buffer [ (option [ , option] ) ] ( loadstatement | selectstatement )
```

where

```
option ::= incremental | expiry  
expiry ::= stale [after] amount [(days | hours)]
```

amount is a number specifying the time period. Decimals may be used. The unit is assumed to be **days** if omitted.

The **incremental** option enables the ability to read only part on an underlying file. Previous size of the file is stored in the XML header in the QVD file. This is particularly useful with log files. All records loaded at a previous occasion are read from the QVD file whereas the following new records are read from the original source and finally an updated QVD file is created.

Note The **incremental** option can only be used with text files. It cannot be used for incremental load if previously read data has been changed or deleted from the source file.

The **stale after** option is typically used with DB sources where there is no simple timestamp on the original data. Instead one specifies how old the QVD snapshot can be to be used. A **stale after** clause simply states a time period from the creation time of the buffer that will be used as source for data and after that the original data source will be used. The QVD buffer file will then automatically be updated and a new period starts.

If no option is used, the QVD buffer created by the first execution of the script will be used indefinitely.

Examples:

```
buffer select * from MyTable;
buffer (stale after 7 days) select * from MyTable;
buffer (incremental) load * from MyLog.log;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Bundle

If a piece of external information, such as an image or a sound is to be connected to a field value, this can be done via a table that is loaded with the **info** prefix (see page 302 for details). If you want to include the external files into the qvw itself, e.g for portability reasons, you may use the **bundle** prefix. The info files will be stored in a compressed way but will nevertheless take up space both in the file and in RAM. It is therefore not recommended to use the **bundle** prefix when files are too large or too many.

The info may be referenced from the layout as normal info, via the info chart function or referenced as an internal file via the special syntax **qmem://fieldname | fieldvalue** alternatively **qmem://fieldname | < index >** where *index* is the internal index of a field value.

The **info** prefix may be omitted when **bundle** is used.

The syntax is:

```
bundle [info] ( loadstatement | selectstatement )
```

Examples:

```
Bundle info Load * from flagoced.csv;
Bundle Select * from infotable;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Call

The **call** control statement calls a subroutine which must be defined by a **sub** statement.

The syntax is:

call *name* ([*paramlist*])

where

name is the name of the subroutine.

paramlist is a comma separated list of the actual parameters to be sent to the subroutine. Each item in the list may be a field name, a variable name or an arbitrary expression.

The subroutine called by a **call** statement must be defined by a **sub** statement encountered earlier during script execution.

Parameters are copied into the subroutine and, if the parameter in the **call** statement is a variable name, copied back out again upon exit from the subroutine.

Since the **call** statement is a control statement it must not cross a line boundary. It may be terminated by semicolon or end-of-line.

Examples:

```

// example 1
sub INCR (I,J)
  I = I + 1
  exit sub when I > 10
  J = J + 1
end sub
call INCR (X,Y)

// example 2 - list all QV related files on disk
sub DoDir (Root)
  for each Ext in 'qvw','qvo','qvs','qvt','qvd'
    for each File in filelist (Root&'\*.'&Ext)
      Load '$(File)' as Name,
      FileSize('$ (File)') as Size,
      FileTime('$ (File)') as FileTime
      autogenerate 1;
    next File
  next Ext
  for each Dir in dirlist (Root&'\*')
    call DoDir (Dir)
  next Dir
end sub
call DoDir ('C:')

```

Concatenate

If two tables that are to be concatenated have different sets of fields, concatenation of two tables can be forced with the **concatenate** prefix. This statement forces concatenation with an existing named table or the latest previously created logical table. See further page 467.

The syntax is:

concatenate [(tablename)] (loadstatement | selectstatement)

Examples:

```

Load * from file1.csv;
Concatenate load * from file2.csv;
Concatenate select * from table3;

tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
. . . .
Concatenate (tab1) load * from file3.csv;

```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Connect

A general database can be accessed through the OLE DB/ODBC interface, but first, the data source must be specified. This is done through the **connect** statement. The syntax is:

```
[ODBC] connect to connect-string [ ( access_info ) ]
```

where

```
connect-string ::= datasourcename { ; conn-spec-item }
```

The connection string is the name of the data source and an optional list of one or more connection specification items. If the name of the data source contains blanks, or if any connection specification items are listed, the connection string must be enclosed by quotation marks.

datasourcename must be a defined ODBC data source.

```
conn-spec-item ::= DBQ= database_specifier | DriverID=
driver_specifier | UID= userid | PWD= password
```

The possible connection specification items may differ between different databases. For some databases, other items than those above are also possible. Normally only the database specifier (DBQ) is used.

```
access_info ::= access_item { , access_item }
```

```
access_item ::= (userid is username | xuserid is
scrambledusername ) | (password is password | xpassword is
scrambledpassword ) | codepage is codepageID | mode is write
```

```
codepageID ::= ansi | oem | codepage number
```

If the **connect** statement is generated by the provided wizard any user ID and password provided will be generated with the scrambled **xuserid is / xpassword is** syntax, provided that **Scramble Connect User Credentials** is selected on the **General** page of the **User Preferences** dialog (see page 91). If you type the connect statement yourself the non-scrambled **userid is / password is** syntax must be used for providing user ID and password. Full scrambling is currently only possible for **ODBC connect** statements. Some parts of the **OLEDB connect** string cannot be scrambled.

The **codepage is** specifier can be used if you encounter problems with national characters in specific ODBC/OLE DB drivers.

If **mode is write** is specified in the *access_info* the connection will be opened in read-write mode. In all other cases the connection will be opened as read-only. The use of **mode is write** must be enabled in the **Edit Script** dialog (see page 217).

If **ODBC** is placed before **connect**, the ODBC interface will be used; else, OLE DB will be used.

Example:

```
Connect to 'Nwind;
DBQ=C:\ProgramFiles\MSOffice95\Access\Samples\Northwi
nd.mdb' (UserID is sa, Password is admin);
```

The data source defined through this statement is used by subsequent **select** statements, until a new **connect** statement is made.

Crosstable

To turn a cross table into a straight table, you can use a **crosstable** prefix. For a description of cross tables and how they can be handled in QlikView see page 476.

The syntax is:

```
crosstable ( attribute field , data field [ , n ] )
( loadstatement | selectstatement )
```

where

attribute field is the field to contain the attribute values.

data field is the field to contain the data values.

n is the number of qualifier fields preceding the table to be transformed to generic form. Default is 1.

Examples:

```
Crosstable (Month,Sales) Load * from ex1.csv;
Crosstable (Month,Sales,2) Load * from ex2.csv;
Crosstable (A,B) Select * from table3;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Directory

Statements accessing the computer's file system, e.g. the **load** statement, can be preceded by a **directory** statement, defining in what directory to search for the files.

The syntax is:

```
directory path
```

where

path is a text that can be interpreted as the path to the file, either absolute, or relative to the qvw file.

Example:

```
Directory c:\userfiles\data;
```

The directory defined through this statement is used by subsequent **load** statements until a new **directory** statement is made.

A **directory** statement is generated automatically if two or more files are selected simultaneously in the **Open Local Files** dialog.

Disconnect

Terminates the current ODBC/OLEDB connection. This statement is optional. The connection will be automatically terminated when a new **connect** statement is executed or when the script execution is finished. The syntax is:

```
disconnect
```

Example:

```
disconnect;
```

Do..Loop

The **do..loop** control statement creates a loop which will be executed until a logical condition is met.

The syntax is:

```
do [ ( while | until ) condition ]  
    [ statements ]  
[ exit do [ ( when | unless ) condition ]  
    [ statements ] ]  
loop [ ( while | until ) condition ]
```

where

condition is a logical expression evaluating to true or false.

statements is any group of one or more QlikView script statements.

The **while** or **until** conditional clause must only appear once in any **do..loop** statement, i.e. either after **do** or after **loop**. Each *condition* is interpreted only the first time it is encountered but is evaluated for every time it is encountered in the loop.

If an **exit do** clause is encountered inside the loop, the execution of the script will be transferred to the first statement after the **loop** clause denoting the

end of the loop. An **exit do** clause can be made conditional by the optional use of a **when** or **unless** suffix.

Since the **do..loop** statement is a control statement, each of its three possible clauses (**do**, **exit do** and **loop**) must not cross a line boundary. They may be terminated by semicolon or end-of-line.

Examples:

```
// load all files file1.csv..file9.csv
let a=1;
do while a<10
    load * from files$(a).csv;
    let a=a+1;
loop
```

Drop Field

One or more QlikView fields can be dropped from memory at any time during script execution by using a **drop field** statement.

The syntax is:

```
drop field[s] fieldname [ , fieldname2 ... ] [from tablename1 [ , tablename2 ... ]]
```

Note Both **drop field** and **drop fields** are allowed forms with no difference in effect. If no table is specified, the field will be dropped from all tables where it occurs.

Examples:

```
drop field A;
drop fields A,B;
drop field A from X;
drop fields A,B from X,Y;
```

Drop Table

One or more QlikView input tables can be dropped from memory at any time during script execution by using a **drop table** statement. The following will disappear as a result of such a statement:

- the table(s)
- all fields which are not part of remaining tables
- field values in remaining fields, which came exclusively from the dropped table(s)

The syntax is:

drop table[s] *tablename [, tablename2 ...]*

Note Both **drop table** and **drop tables** are allowed forms with no difference in effect.

Examples:

```
drop table Orders, Salesmen, T456a;
// drops three tables
Tab1:
Select * from Trans;
Load Month, Sum(Sales) resident Tab1 group by Month;
Drop table Tab1;
// Only the aggregates remain in memory.
// Trans data discarded.
```

Execute

With the **execute** statement it is possible to execute other programs during the execution of the QlikView script. The QlikView script execution is temporarily halted waiting for the other application to finish. The use of the **execute** statement must be enabled in the **Edit Script** dialog (see page 217).

The syntax is:

execute *commandline*

where

commandline is a text that the operating system can interpret as a command line.

Examples:

```
Execute C:\MSOffice95\Excel\Excel.exe;
Execute winword macro.doc;
Execute cmd.exe /C C:\BatFiles\Log.bat
```

Note The /C needs to be included as it is a parameter to cmd.exe.

Exit Script

The **exit script** control statement stops script execution. It may be inserted anywhere in the script. An **exit script** statement can be made conditional by the optional use of a **when** or **unless** clause.

The syntax is:

exit script [(**when** | **unless**) *condition*]

where

condition is a logical expression evaluating to true or false.

Since the **exit script** statement is a control statement it must not cross a line boundary. The statement may be terminated by semicolon or end-of-line.

Examples:

```
exit script
exit script;
exit script when A=1
```

First

The **first** prefix to a **load** or **select** statement is used to load only a set maximum number of records from the input table, regardless of its length.

The syntax is:

```
first n ( loadstatement | selectstatement )
```

where

n is an arbitrary expression which evaluates to an integer indicating the maximum number of records to be read.

Examples:

```
First 10 Load * from abc.csv;
First 1 Select * from Orders;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

For..Next

The **for..next** control statement creates a loop with a counter. The statements inside the loop enclosed by **for** and **next** will be executed for each value of the counter variable between specified low and high limits.

The syntax is:

```
for counter = expr1 to expr2 [ step expr3 ]
  [ statements ]
[ exit for [ ( when | unless ) condition ]
  [ statements ] ]
next [ counter ]
```

where

counter is a script variable name. If *counter* is specified after **next** it must be the same variable name as the one found after the corresponding **for**.

expr1 is an expression which evaluates to the first value of the *counter* variable for which the loop should be executed.

expr2 is an expression which evaluates to the last value of the *counter* variable for which the loop should be executed.

expr3 is an expression which evaluates to a value indicating the increment of the *counter* variable each time the loop has been executed.

condition is logical expression evaluating to true or false.

statements is any group of one or more QlikView script statements.

The expressions *expr1*, *expr2* and *expr3* are only evaluated the first time the loop is entered. The value of the *counter* variable may be changed by statements inside the loop, but such practice normally makes a script more difficult to follow.

If an **exit for** clause is encountered inside the loop, the execution of the script will be transferred to the first statement after the **next** clause denoting the end of the loop. An **exit for** clause can be made conditional by the optional use of a **when** or **unless** suffix.

Since the **for..next** statement is a control statement, each of its three possible clauses (**for..to..step**, **exit for** and **next**) must not cross a line boundary. They may be terminated by semicolon or end-of-line.

Examples:

```
// load files file1.csv..file9.csv
for a=1 to 9
    load * from file$(a).csv;
next

for counter=1 to 9 step 2
    set filename=X$(counter).csv;
    if rand()<0.5 then
        exit for unless counter=1
    end if
    load a,b from $(filename);
next
```

For Each..Next

The **for each..next** control statement creates a loop which executes for each value in a comma separated list. The statements inside the loop enclosed by **for** and **next** will be executed for each value of the list. Special syntax makes it possible to generate lists with file and directory names in the current directory.

The syntax is:


```

for each var in list
    [ statements ]
[ exit for [ ( when | unless ) condition ]
    [ statements ] ]
next [ var ]

```

where

var is a script variable name which will acquire a new value from *list* for every loop execution. If *var* is specified after **next** it must be the same variable name as the one found after the corresponding **for each**.

list := *item* { , *item* }

item := *constant* | (*expression*) | **filelist** *mask* | **dirlist** *mask*

constant is any number or string.

expression is an arbitrary expression.

mask is a filename or directory name mask which may include any valid filename characters and the standard wildcard characters * and ?.

condition is logical expression evaluating to true or false.

statements is any group of one or more QlikView script statements.

The **filelist** *mask* syntax produces a comma separated list of all files in the current directory matching the filename mask. The **dirlist** *mask* syntax produces a comma separated list of all directories in the current directory matching the directory name mask.

The value of the *var* variable may be changed by statements inside the loop, but such practice normally makes a script more difficult to follow.

If an **exit for** clause is encountered inside the loop, the execution of the script will be transferred to the first statement after the **next** clause denoting the end of the loop. An **exit for** clause can be made conditional by the optional use of a **when** or **unless** suffix.

Since the **for each..next** statement is a control statement, each of its three possible clauses (**for each**, **exit for** and **next**) must not cross a line boundary. They may be terminated by semicolon or end-of-line.

Examples:

```
for each a in 1,3,7,'xyz'
  load * from file$(a).csv
next

// list all QV related files on disk
sub DoDir (Root)
  for each Ext in 'qvw','qva','qvo','qvs'
    for each File in filelist (Root&'\'&Ext)
      Load '$(File)' as Name,
      FileSize('$ (File)') as Size,
      FileTime('$ (File)') as FileTime
      autogenerate 1;
    next File
  next Ext
  for each Dir in dirlist (Root&'\'*')
    call DoDir (Dir)
  next Dir
end sub
call DoDir ('C:')
```

Force

The **force** statement forces QlikView to interpret field values of subsequent **load** and **select** statements as written with only upper case letters, with only lower case letters, always capitalized or as they appear (mixed). With the help of this statement it is possible to associate field values of tables made according to different conventions.

The syntax is:

```
force ( capitalization | case upper | case lower | case mixed )
```

Examples:

```
Force Capitalization;
Force Case Upper;
Force Case Lower;
Force Case Mixed;
```

If nothing is specified, force case mixed is assumed. The **force** statement is valid until a new **force** statement is made.

The **force** statement has no effect in the access section: all field values loaded are case insensitive.

Generic

The unpacking and loading of a generic database can be done with a **generic** prefix. For a description of a generic database, see page 475.

The syntax is:

```
generic ( loadstatement | selectstatement )
```

Examples:

```
Generic Load * from abc.csv;
Generic Select * from table1;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Tables loaded through a **generic** statement cannot be concatenated.

Hierarchy

The **hierarchy** keyword is a prefix to the **load** and **select** statements. It is used to transform an adjacent nodes table to an expanded nodes table.

Hence, the input table must be an adjacent nodes table as described in section Hierarchies, page 481. The output of the **hierarchy** prefix is an expanded nodes table with all the needed additional fields.

Normally, the input table has exactly one record per node and in such a case the output table will contain the same number of records. However, sometimes there are nodes with multiple parents, i.e. one node is represented by several records. If so, the output table may have more records than the input table.

All nodes without a parent id or with a parent id not found in the nodeID column will be considered as roots. Also, only nodes with a connection to a root node – direct or indirect – will be loaded, thus avoiding circular references.

Additional fields containing the name of the parent node, the path of the node and the depth of the node can be created.

The syntax is:

```
Hierarchy (NodeID, ParentID, NodeName, [ParentName], [PathSource], [PathName], [PathDelimiter], [Depth]) (loadstatement | selectstatement)
```

Where

NodeID is the name of the field that contains the node id.

ParentID is the name of the field that contains the node id of the parent node.

NodeName is the name of the field that contains the name of the node.

ParentName is a string used to name the new **ParentName** field. If omitted, this field will not be created.

PathSource is the name of the field that contains the name of the node used to build the node path. Optional parameter. If omitted, **NodeName** will be used.

PathName is a string used to name the new **Path** field, which contains the path from the root to the node. Optional parameter. If omitted, this field will not be created.

PathDelimiter is a string used as delimiter in the new **Path** field. Optional parameter. If omitted, '/' will be used.

Depth is a string used to name the new **Depth** field, which contains the depth of the node in the hierarchy. Optional parameter. If omitted, this field will not be created.

Example:

```
Hierarchy(NodeID, ParentID, NodeName) Load
NodeID,
ParentID,
NodeName,
Attribute
From data.xls (biff, embedded labels, table is
[Sheet1$]);
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

HierarchyBelongsTo

The **hierarchybelongssto** keyword is a prefix to the **load** and **select** statements. The input table must be an adjacent nodes table as described in section Hierarchies., page 481.

The prefix creates a table containing all ancestor-child relations of the hierarchy. The ancestor fields can then be used to select entire trees in the hierarchy. The output table usually contains several records per node.

An additional field containing the depth difference of the nodes can be created.

The syntax is:

HierarchyBelongsTo (*NodeID, ParentID, NodeName, AncestorID, AncestorName, [DepthDiff]*) (*loadstatement | selectstatement*)

Where

NodeID is the name of the field that contains the node id.

ParentID is the name of the field that contains the node id of the parent node.

NodeName is the name of the field that contains the name of the node.

AncestorID is a string used to name the new ancestor id field, which contains the id of the ancestor node.

AncestorName is a string used to name the new ancestor field, which contains the name of the ancestor node.

DepthDiff is a string used to name the new **DepthDiff** field, which contains the depth of the node in the hierarchy relative the ancestor node. Optional parameter. If omitted, this field will not be created.

Example:

```
HierarchyBelongsTo (NodeID, ParentID, Node, Tree,
ParentName) Load
NodeID,
ParentID,
NodeName
From data.xls (biff, embedded labels, table is
[Sheet1$]);
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

If..Then..Elseif..Else..End If

The **if..then** control statement creates a conditional clause, which causes the script execution to take different paths depending on one or more logical conditions.

The syntax is:

```
if condition then
    [ statements ]
{ elseif condition then
    [ statements ] }
[ else
    [ statements ] ]
end if
```

where

condition is a logical expression which can be evaluated as true or false.

statements is any group of one or more QlikView script statements.

Since the **if..then** statement is a control statement each of its four possible clauses (**if..then**, **elseif..then**, **else** and **end if**) must not

cross a line boundary. They may be terminated by semicolon or end-of-line.

Examples:

```
if a=1 then
    load * from abc.csv;
    sql select e,f,g from tabl;
end if

if a=1 then; drop table xyz; end if;

if x>0 then
    load * from pos.csv;
elseif x<0 then
    load * from neg.csv;
else
    load * from zero.csv;
end if
```

Image_size

This statement is used with **Info** to resize images from a database management system to fit in the fields. Width and height is specified in pixels.

Example:

```
Info Image_size(122,122) Select ID, Photo From
infotable;
```

Info

If a piece of information, such as a text file, an image or a multimedia presentation is to be connected to a field value, this is done in a table that is loaded through an **info** prefix. The table must contain two columns only, the first one with the field values that should be keys to the information, the second one containing the file names of the images etc. See further page 487. Information can also be embedded inside a QlikView document by using the bundle prefix, see page 287.

The same applies to, for example, a picture from a database management system. On a binary field, a blob, the **info select** statement makes an implicit **bundle**, i.e. the binary data will be fetched immediately and stored in the qvw. The binary data must be the second field in a select statement. The syntax is:

```
info ( loadstatement | selectstatement )
```

Examples:

```
Info Load * from flagoecd.csv;
Info Select * from infotable;
Info Select Key, Picture From infotable;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Info can also be embedded inside a QlikView document. See the **bundle** prefix syntax on page 287.

Inner

The **join** and **keep** prefixes in the QlikView script language can be preceded by the prefix **inner**.

If used before **join**, it specifies that the join between the two tables should be an inner join. The resulting table contains only combinations between the two tables with a full data set from both sides.

If used before **keep**, it specifies that the two tables should be reduced to their common intersection before being stored in QlikView. See further on page 469.

The syntax is:

```
inner (join / keep) [(tablename)] ( loadstatement |selectstatement )
```

Examples:

```
QVTable:
Select * from Table1;
inner join select * from Table2;
```

```
QVTab1:
Select * from Table1;
QVTab2:
inner keep select * from Table2;
```

```
tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
. . . .
inner keep (tab1) load * from file3.csv;
```

The **outer** (page 323), **left** (page 307) and **right** (page 328) prefixes can be used in similar ways.

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Inputfield

A field may be flagged as an input field by listing it in an **inputfield** statement before it is referenced in any **load** or **select** statement.

Input fields behave somewhat differently as compared to regular fields. The most important difference is their ability to accept changes in field values, interactively or programmatically, without running the script. Field values must be loaded into the field via **load** or **select** statements. Each field value loaded in the script will create a placeholder for a field value replacement value. Only existing field values can thus be changed interactively or programmatically. In input fields all field values will be treated as distinct, regardless if several have the same value.

The syntax is

```
inputfield ListOfFields
```

where

ListOfFields is one or more field names separated by commas.

The wildcard characters * and ? are allowed in field names. Quoting of field names is sometimes necessary when using the wildcards as part of field names.

Examples:

```
Inputfield B;  
Inputfield A,B;  
Inputfield B??x*;
```

IntervalMatch

The **intervalmatch** prefix to a **load** or **select** statement is used to link discrete numeric values to one or more numeric intervals. See further page 477.

The syntax is:

```
intervalmatch ( matchfield ) ( loadstatement | selectstatement )
```

where

matchfield is the field containing the discrete numeric values to be linked to intervals.

Loadstatement | *selectstatement* must result in a two-column table, where the first field contains the lower limit of each interval and the second field contains the upper limit of each interval. The intervals are always closed, i.e. the end points are included in the interval. Non-numeric limits render the interval to be disregarded.

The intervals may be overlapping and the discrete values will be linked to all matching intervals.

Examples:

```
intervalmatch (Day) Load PeriodStart, PeriodEnd from
intervals.csv;
```

```
intervalmatch (Day) Select PeriodStart, PeriodEnd from
intervaltable;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329. The extended **intervalmatch** statement syntax is described below and on page 479.

IntervalMatch (Extended Syntax)

The extended **intervalmatch** prefix to a **load** or **select** statement is used to match discrete data to one or more dimensions that are changing over time. This is a very powerful and flexible feature which can be used to resolve problems with slowly changing dimensions by linking the specific key-field(s) to the appropriate numerical interval.

The syntax is:

```
intervalmatch (matchfield, keyfield1 [, keyfield2,...keyfield5])
(loadstatement | selectstatement )
```

where *matchfield* is the field containing the discrete numeric values to be linked to the interval; and the *keyfield(s)* are fields that contain the attributes that are to be matched using the interval.

Loadstatement | *selectstatement* must result in a table where the first two fields contain the lower and upper limits of each interval and the third (and following fields) contains the keyfield(s) present in the intervalmatch statement. The intervals are always closed, i.e. the end points are included in the interval. Null limits render the interval to be disregarded (undefined) while non-numeric limits extend the interval infinitely (unlimited). The intervals may be overlapping and values will be linked to all matching intervals.

In order to avoid that undefined interval limits are disregarded it is necessary to allow null values to map to the fields that constitute the lower or upper limits of the interval. This is done by using the **nullasvalue** statement described on page 321.

Before the **intervalmatch** statement, the field containing the discrete data points (*Date* in the example below) must already have been read into QlikView. The **intervalmatch** statement does not by itself read this field from the database table.

Examples:

```
inner join intervalmatch (Date,Key) load FirstDate,  
LastDate, Key resident Key;
```

On page 479 you will find comments on the use of the extended **interval-Match** statement.

Note The inner join prefix that precedes the **intervalmatch** prefix in the example above is required in order to combine the data from the interval table with the data from the match table into one table.

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Join

The **join** prefix joins the loaded table with an existing named table or the last previously created logical table. The join is a natural join made over all the common fields. See further page 467. The **join** prefix may be preceded by one of the prefixes **inner** (page 303), **outer** (page 323), **left** (page 307) or **right** (page 328).

The syntax is:

```
join [(tablename)] ( loadstatement | selectstatement )
```

Examples:

```
Join Load * from abc.csv;  
Join Select * from table1;  
  
tab1:  
Load * from file1.csv;  
tab2:  
load * from file2.csv;  
.. .. ..  
join (tab1) load * from file3.csv;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Keep

The **keep** prefix between two **load** or **select** statements has the effect of reducing one or both of the two tables before they are stored in QlikView based on the intersection of table data. The **keep** prefix must always be preceded by one of the prefixes **inner** (page 303), **left** (page 307) or **right** (page 328). The selection of records from the tables is made in the same way

as in a corresponding join. However, the two tables are not joined and will be stored in QlikView as two separately named tables. See further page 467.

The syntax is:

```
keep [ (tablename) ] ( loadstatement | selectstatement )
```

Examples:

```
Left Keep Load * from abc.csv;
Inner Keep Select * from table1;

tab1:
Load * from file1.csv;

tab2:
load * from file2.csv;
. . . .
left keep (tab1) load * from file3.csv;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Left

The **join** and **keep** prefixes in the QlikView script language can be preceded by the prefix **left**.

If used before **join**, it specifies that the join between the two tables should be a left join. The resulting table only contains combinations between the two tables with a full data set from the first table.

If used before **keep**, it specifies that the second table should be reduced to its common intersection with the first table before being stored in QlikView. See further page 467.

The syntax is:

```
left ( join / keep ) [(tablename)]( loadstatement | selectstatement )
```

Examples:

```
QVTable:
Select * from Table1;
left join select * from Table2;

QVTab1:
Select * from Table1;
QVTab2:
left keep select * from Table2;

tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
. . . .
left keep (tab1) load * from file3.csv;
```

The **inner** (page 303), **outer** (page 323), and **right** (page 328) prefixes can be used in similar ways.

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Let

The **let** statement has been created as a complement to the **set** statement (described on page 332), used for defining script variables. The **let** statement, in opposition to the **set** statement, evaluates the expression on the right side of the '=' before it is assigned to the macro variable.

The word **let** may be omitted, but the statement then becomes a control statement. Such a statement without the keyword **let** must be contained within a single script row and may be terminated either with a semicolon or the end-of-line.

The syntax is:

```
[ let ] variablename = expression
```

Examples:

```
Set X=3+4;
Let Y=3+4;
Y=3+4
Z=$((Y)+1);
Let T=now();
```

\$(X) will be evaluated as '3+4', whereas **\$(Y)** will be evaluated as '7'.

\$(Z) will be evaluated as '8'.

\$ (T) will be given the value of the current time.

Load

The **load** statement loads fields from a typed ASCII file, from data defined in the script, from a previously loaded input table, from a previously loaded field, from the result of a following **select** statement, or by generating data automatically. The general syntax of the **load** statement is:

```
load [ distinct ] *fieldlist
      [( from file [ format-spec ] | from_field field [format-spec] |
      inline [ format-spec ] data |
      resident table-label |
      autogenerate size)]
      [ where criterion | while criterion ]
      [ group by fieldlist ]
      [ order by field [sortorder] { , field [sortorder] } ]
```

where

distinct is a predicate used if only the first of duplicate records should be loaded.

**fieldlist*::= (* | *field*) { , *field* })

A list of the fields to be selected. The " * " is used to select all the fields in the table.

field::= (*fieldref* | *expression*) [**as** *aliasname*]

The field definition must always contain a reference to an existing field, or an expression.

fieldref::= (*fieldname* | @*fieldnumber* | @*startpos:endpos*[**I** | **U** | **R** | **B**])

fieldname is a text that is identical to a field name in the table. Note that the field name must be enclosed by straight double quotation marks or square brackets if it contains e.g. spaces. Sometimes field names are not explicitly available. Then a different notation is used:

@*fieldnumber* represents the field number in a delimited text file. It must be a positive integer preceded by "@". The numbering is always made from 1 and up to the number of fields.

@*startpos:endpos* represents the start and end positions of a field in a file with fixed length records. The positions must both be positive integers. The two numbers must be pre-

ceded by "@" and separated by a colon. The numbering is always made from 1 and up to the number of positions. If *@startpos:endpos* is immediately followed by the characters **I** or **U**, the bytes read will be interpreted as a binary signed (**I**) or unsigned (**U**) integer (Intel byte order). The number of positions read must be 1, 2 or 4. If *@startpos:endpos* is immediately followed by the character **R**, the bytes read will be interpreted as a binary real number (IEEE 32-bit or 64-bit floating point). The number of positions read must be 4 or 8. If *@startpos:endpos* is immediately followed by the character **B**, the bytes read will be interpreted as a binary real number (Binary Coded Decimal) according to the COMP-3 standard. Any number of bytes may be specified.

expression can be a numeric function or a string function based on one or several other fields in the same table. For the syntax of expressions, see page 355.

as is used to assign a new name to the field.

from is used if data should be loaded from a file, and not typed within the script.

file::= [path] filename

The *path* is the path to the file, either absolute, or relative to the QlikView document (qvw file). If the path is omitted, QlikView searches for the file in the directory specified by the **directory** statement. If there is no **directory** statement, QlikView searches in the working directory, which is usually the directory in which the QlikView file is located. The path may also be a URL address (HTTP or FTP), pointing to a location on the Internet or an intranet.

The *filename* may contain the standard DOS wildcard characters (* and ?). This will cause all the matching files in the specified directory to be loaded.

format-spec ::= (fspec-item { , fspec-item })

The format specification consists of a list of several format specification items, within brackets. For a description of these, see below, page 314.

from_field is used if data should be loaded from a previously loaded field.

field::= (tablename, fieldname)

The field is the name of the previously loaded tablename and fieldname.

inline is used if data should be typed within the script, and not loaded from a file. Use the **Inline Data Wizard** for help with the creation of **load inline** statements, see page 241.

data::= [text]

Data entered through an **inline** clause must be enclosed by quotation marks or square brackets. The text between these is interpreted in the same way as the content of a file.

Hence, where you would insert a new line in a text file, you should also do it in the text of an inline clause, i.e. by pressing the ENTER key when typing the script.

resident is used if data should be loaded from a previously loaded input table.

Table label (see page 343) is a label preceding the **load** or **select** statement(s) that created the original input table. The label shall be given with a colon at the end.

autogenerate is used if data should be automatically generated by QlikView.

size::=number

Number is an integer indicating the number of records to be generated. The field-list must not contain expressions that require data from a database. Only constants and parameter-free functions are allowed in the expressions.

where is a clause used to state whether a record should be included in the selection or not. The selection is included if *criterion* is true. See below under *Examples, Selecting certain records*.

while is a clause used to state whether a record should be repeatedly read. The same record is read as long as *criterion* is true. In order to be useful, a **while** clause must typically include the **IterNo()** function (see description of QlikView functions, page 402).

criterion is a logical expression. For the syntax of expressions, see page 355.

fieldname, path, filename and *aliasname* are text strings representing what the respective names imply.

group by is a clause used to define over which fields the data should be aggregated (grouped). The aggregation fields should be included in the expressions loaded in some way. No other fields than the aggregation fields may be used outside aggregation functions (see page 359) in the loaded expressions.

fieldlist ::= (field { , field })

order by is a clause used to sort the records of a resident input table before they are processed by the **load** statement. The resident table can be sorted by one or more fields in ascending or descending order. The sorting is made primarily by numeric value and secondarily by national ASCII value. This clause may only be used when the data source is a resident table.

field is the field in the resident table to sort by. The field can be specified by name or by its number in the resident table (the first field is number 1).

sortorder is either **asc** for ascending or **desc** for descending. If no sort order is specified, **asc** is assumed.

Note If no source of data is given by means of a **from**, **inline**, **resident** or **autogenerate** clause, data will be loaded from the result of the immediately succeeding **load** or **select** statement. The succeeding statement should not have a prefix. The result of the succeeding **load** or **select** statement will be treated as a temporary input table, which is lost as soon as its contents have been used for the preceding **load/select** statement.

Examples:

Different file formats:


```
Load * from data1.csv;

Load * from 'c:\userfiles\data1.csv' (ansi, txt,
delimiter is ',', embedded labels);

Load * from 'c:\userfiles\data2.txt' (ansi, txt,
delimiter is '\t', embedded labels);

Load * from file2.dif (ansi, dif, embedded labels);

Load @1:2 as ID, @3:25 as Name, @57:80 as City from
data4.fix (ansi, fix, no labels, header is 0, record
is 80);
```

Selecting certain fields, calculating fields

```
Load FirstName, LastName, Number from data1.csv;
Load @1 as A, @2 as B from data3.txt (ansi, txt,
delimiter is '\t', no labels);

Load FirstName&' '&LastName as Name from data1.csv;

Load Quantity, Price, Quantity*Price as Value from
data1.csv;
```

Selecting certain records

```
Load distinct FirstName, LastName, Number from
data1.csv;

Load * from Consumption.csv where Litres>0;

Load * from Consumption.csv where Litres>0 and
Litres<1000;
```

Loading data not on file

```
Load * Inline
[CatID, Category
0,Regular
1,Occasional
2,Permanent];
```

```

Load * Inline
[UserID, Password, Access
A, ABC456, User
B, VIP789, Admin];

Load RecNo() as A, rand() as B autogenerate(10000);

```

Loading data from previously loaded input table

```

tabl:
Select A,B,C,D from transtable;
Load A,B,month(C),A*B+D as E resident tabl;
Load A,A+B+C resident tabl where A>B;
Load A,B*C as E resident tabl order by A;
Load A,B*C as E resident tabl order by 1,2;
Load A,B*C as E resident tabl order by C desc, B asc,
1 desc;

```

Loading data from previously loaded fields

```

Load A from_field (Characters, Types);

```

Loading data from succeeding input table

```

Load A, B, if(C>0,'+', '-') as X, weekday(D) as Y;
Select A,B,C,D from Table1;

```

Grouping data

```

Load ArtNo, round(Sum(TransAmount),0.05) as
ArtNoTotal from table.csv group by ArtNo;

Load Week, ArtNo, round(Avg(TransAmount),0.05) as
WeekArtNoAverages from
table.csv group by Week, ArtNo;

```

Reading one record repeatedly

```

My Tab:
Load Student,
mid(Grades,IterNo(),1) as Grade,
pick(IterNo(),'Math', 'English',
'Science','History') as Subject
from Tab1.csv
while mid(Grades,IterNo(),1)<>' ';

```

Description of format specification items

Each format specification item defines a certain property of the text file:

fspec-item ::= [ansi | oem | mac | UTF-8 | Unicode / codepage is |

txt | **fix** | **dif** | **biff** / **ooxml** | **html** | **xml** | **qvd** | **xmlsax** | **xmlsimple**
 | **delimiter is** *char* | **no eof** | **embedded labels** | **explicit labels** | **no**
labels | **table is** [*tablename*] | **header is** *n* | **header is line** | **header**
is *n* **lines** | **comment is** *string* / **pattern is** *path* | **record is** *n* |
record is line | **record is** *n* **lines** | **no quotes** | **msq** | **filters** (*filter*
specifiers)]

Character set

ansi
oem
mac
UTF-8
Unicode

The file can be written with the **ansi** (Windows), the **oem** (DOS, OS/2 and others), **UTF-8**, **Unicode** or the **mac** character set. The conversion from the **oem** character set is not implemented for MacOS. If nothing is specified, **ansi** is assumed under Windows and **mac** under MacOS.

Note Under Windows 95/98/ME, files with the **mac** character set will be interpreted as **ansi**. In order to get correct interpretation of national character an applicable nls-file (national language support) must be installed (see Microsoft knowledge base article Q230147) and instead of the **mac** specifier an explicit code page must be stated in the load statement.

Example:

```
load * from a.txt (codepage is 10000, txt, delimiter
is ',', no labels)
```

Table format

txt
fix
dif
biff
ooxml
html
qvd
xml
xmlsax

In a delimited text file, **txt**, the columns in the table are separated by a character. In a fixed record length file, **fix**, each column is a certain number of characters wide. In a **dif** file, (Data Interchange Format) a

special format for defining the table is used. **biff** (Binary Interchange File Format) is the format used by MS Excel .xls files. **ooxml** (Open Office XML) is used by Microsoft Excel 2007. QVD denotes a file in the proprietary QVD (QlikView Data) file format, exported from a QlikView document. If nothing is specified, a **txt** file is assumed.

Note In order to use `xmlsax` Microsoft's xml parser MSXML 3.0 or higher must be installed on the computer. MSXML is shipped with e.g. Windows XP and MS Internet Explorer 6. It can also be downloaded from the Microsoft home page.

Delimiter (delimited text files only)

delimiter is *char*

For delimited text files, an arbitrary delimiter can be specified through the **delimiter is** specifier. *char* is here a single character with or without quotation marks or a "\t" representing a tab sign, or "\\" representing a backslash character, or the word spaces representing all combinations of one or more spaces. If nothing is specified, **delimiter is ','** is assumed. This specifier is relevant only for **txt** files.

no eof

If this option is used ASCII character 26 which often denotes end-of-file is disregarded.

Labels (field names)

embedded labels

explicit labels

no labels

The field names can be found in different places of the file. If the first record contains the field names, **embedded labels** should be used. If there are no field names to be found, **no labels** should be used. In **dif** files, a separate header section with explicit field names is sometimes used. In such a case, **explicit labels** should be used. If nothing is specified, **embedded labels** is assumed, also for **dif** files.

Table (biff and HTML files only)

table is [*tablename* | @ *tablenumber*]

When loading Excel files, you must specify the table in the workbook. When loading tables from HTML pages you must specify the table number.

Header size (fixed record files, delimited text files and Excel files only)

header is *n*
header is line
header is *n lines*

For fixed record length files, delimited text files and Excel files, an arbitrary header length can be specified through the **header is** specifier. A header is a text section not used by QlikView. The header length can be given in bytes (**header is** *n*), or in lines (**header is line** or **header is** *n lines*). *n* is here a positive integer, representing the header length. If not specified, **header is 0** is assumed.

Comment (fixed record and delimited text files only)

comment is *string*

Records in the file starting with *string* will be disregarded.

Record length (fixed record files only)

record is *n*
record is line
record is *n lines*

For fixed record length files, the record length must be specified through the **record is** specifier. The record length can be given in bytes (**record is** *n*), or in lines (**record is line** or **record is** *n lines*). *n* is here a positive integer, representing the record length. The **record is** specifier is only relevant for **fix** files.

Quotes (text files only)

no quotes
msq

If no quotes are to be accepted in the text file, the **no quotes** specifier should be used. By default, the quotes " " or ' ' can be used, but only if they are the first and last non blank character of a field value. The **no quotes** specifier is only relevant for **txt** files. **msq** indicates that modern style quoting is assumed.

XML (xml files only)

xmlsax
xmlsimple
pattern is *path*

xmlsax and **xmlsimple** are mutually exclusive, only one can be specified when using xml. When using **pattern** the file will be read from the start of the specified tag to the end of the tag. If *path* contains spaces it must be quoted.

Filters

filter specifiers ::= (filterspec { , filterspec })

Loosen Table

One or more QlikView input tables can be explicitly declared loosely coupled (see chapter “Loosely Coupled Tables” on page 144 for details on loosely coupled tables) during script execution by using a **loosen table** statement.

Tables declared as loosely coupled by this script statement will behave somewhat differently from those made loosely coupled interactively. The Loosen Table script statement will prevent the formation of synthetic keys in the table and the loosely coupled status of that table will not be possible to change interactively via the **Table** page of the **Document Properties** dialog.

The syntax is:

```
loosen table[s] tablename [ , tablename2 ...]
```

Both **loosen table** and **loosen tables** are allowed forms with no difference in effect.

Example:

```
Tab1:  
Select * from Trans;  
Loosen table Tab1;
```

Note When QlikView finds circular references in the data structure which cannot be broken by tables declared loosely coupled interactively or explicitly in the script, one or more additional table will be forced loosely coupled until no circular references remain. Tables made loosely coupled in this way have the same status as those made loosely coupled interactively via the **Table** page of the **Document Properties** dialog.

Mapping

The **mapping** prefix is used to create a mapping table. Tables read via **mapping load** or **mapping select** are treated differently from other tables. They will be stored in a separate area of the memory and used only as mapping tables during script execution. After script execution they will be automatically dropped.

A mapping table must have two columns, the first one containing comparison values and the second the desired mapping values. The two columns

must be named, but the names have no relevance in themselves. The column names have no connection to field names in regular input tables.

When mapping tables are used to map a certain field value or expression, that value will be compared to the values in the first column of the mapping table. If found, the original value will be replaced by the corresponding value in the second column of the mapping table. If not found, no replacement is made. See further “Map..Using” on page 319; “Rename Field” on page 325 and “Rename Table” on page 326. See also page 488.

The syntax is:

```
mapping ( load statement | select statement )
```

Examples:

```
mappingtable:
mapping load * from x.csv;
mapping select a,b from map1;
```

```
map1:
mapping load * inline [
x,y
US,USA
U.S.,USA
America,USA
];
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Note **Mapping load** does not support the use of QVD files as input source. The QVD must first be loaded to a table and then a **mapping load** can be made from this table.

Map..Using

This script statement enables mapping of field values in one or more fields as they are encountered in the script after the **map** statement. The fields are mapped using a previously loaded mapping table specified by name (see above). See also page 488.

The automatic mapping will compare the value of the field to the values in the first column of the mapping table. If found, it will replace the original value with the corresponding value in the second column of the mapping table. If not found, the original value is kept.

The automatic mapping is done last in the chain of events that leads up to the field being stored in the QlikView table. This means that the mapping is not done every time a field name is encountered as part of an expression but rather when the value is stored under the field name in the QlikView table. If you need mapping on expression level, you should use the **ApplyMap** function (see page 411).

For the mapping to take place also during partial script execution, the **map..using** statement must be preceded by **add** (see page 284) or **replace** (see page 326).

The syntax is:

```
map fieldlist using mapname
```

where

```
fieldlist := *| ( field { , field } )
```

Using * as fieldlist indicates all fields.

field is the name of a field that should be mapped from this point in the script. The wildcard characters * and ? are allowed in the field names. Quoting of field names may be necessary when using the wildcards.

mapname is the name of a mapping table previously read in a **mapping load** or **mapping select** statement (see “Mapping” on page 318).

Examples:

```
map Country using Cmap;
```

enables mapping of the field *Country* using the mapping table *Cmap*.

```
map A,B,C using X;
```

enables mapping of the fields *A*, *B* and *C* using the mapping table *X*.

```
map * using GenMap;
```

enables mapping of all fields using the mapping table *GenMap*.

```
map "A*", "?B?" using C;
```

enables mapping of fields starting with *A* and three character fields with *B* as second character using the mapping table *C*.

Example of entire script:

```

mappingtable:
mapping load x,y from CountryMap.csv;

map Country using mappingtable;

load Country, City, Customer, Address from
CustomerInfo.csv;

```

To disable mapping of fields, use the **Unmap** statement (page 340).

Noconcatenate

If two tables have the same set of fields and thus would normally be automatically concatenated, you can prevent the concatenation with the **noconcatenate** prefix. This statement prevents concatenation with any existing logical table with the same set of fields.

The syntax is:

```
noconcatenate ( loadstatement / selectstatement )
```

Example:

```

Load A,B from file1.csv;
Noconcatenate load A,B from file2.csv;

```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

NullAsValue

By default QlikView considers null values to be missing or undefined entities. However, certain database contexts imply that null values are to be considered as special values rather than simply missing values. The fact that null values are normally not allowed to link to other null values can be suspended by means of the **nullasvalue** statement. Thus the null values of the specified fields are considered to be unknown rather than undefined when using the **nullasvalue** statement. The **nullasvalue** statement could be considered to supersede the **nulldisplay** variable (see page 346) at field level.

The syntax is:

```
nullasvalue ListOfFields
```

where *ListOfFields* is one or more field names separated by commas.

The wildcard characters * and ? are allowed in the field names. It is also possible to replace *ListOfFields* by the character * to turn **nullasvalue** on for all fields. Quoting of field names is sometimes necessary when using wildcards as part of field names.

Examples:

```
NullAsValue A,B;  
Load A,B from x.csv;  
NullAsValue *;
```

turns **nullasvalue** on for all fields.

```
NullAsValue "*ID";
```

turns **nullasvalue** on for all fields with field names ending with *ID*.

```
NullAsValue "NR*",A;
```

turns **nullasvalue** on for *A* and all fields with field names starting with *NR*.

```
NullAsValue F??;
```

turns **nullasvalue** on for all fields with three character field names starting with *F*.

Please note the following points when using the **nullasvalue** statement:

- Linking of null values is always turned off by default at the beginning of script execution.
- Linking of null values of a specific field can be turned on at any time using a **nullasvalue** statement.
- Linking of null values can be turned off at any time using a **nullasnull** statement (see below).

NullAsNull

The **nullasnull** statement turns off the linking of null values previously set by a **nullasvalue** statement. See further on page 321.

The syntax is:

```
nullasnull ListOfFields
```

where

ListOfFields is one or more field names separated by commas.

The wildcard characters *** and *?* are allowed in the field names. It is also possible to replace *ListOfFields* by the character *** to turn **nullasnull** on for all fields. Quoting of field names is sometimes necessary when using wildcards as part of field names.

Examples:

```
NullAsValue A,B;  
Load A,B from x.csv;  
NullAsNull *;
```

```
NullAsNull *;
    turns NullAsValue off for all fields.

NullAsNull "*ID";
    turns NullAsValue off for all fields with field names ending
    with ID.

NullAsNull "NR*", A;
    turns NullAsValue off for A and all fields with field names
    starting with NR.

NullAsNull F??;
    turns NullAsValue off for all fields with three character
    field names starting with F.
```

Outer

The explicit **join** prefix in the QlikView script language can be preceded by the prefix **outer** in order to specify an outer join. In an outer join all combinations between the two tables are generated. The **outer** prefix is optional. See further page 469.

The syntax is:

```
outer join [(tablename)] ( loadstatement | selectstatement )
```

Example:

```
Select * from Table1;
outer join select * from Table2;
```

is equivalent to:

```
Select * from Table1;
join select * from Table2;
```

The **inner** (page 303), **left** (page 307), and **right** (page 328) prefixes can be used in similar ways.

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Qualify

The automatic join between fields with the same name in different tables can be suspended by means of the **qualify** statement, which qualifies the field name with its table name. If qualified, the field name(s) will be renamed when found in a table. The new name will be in the form of *tablename.fieldname*. *Tablename* is equivalent to the label of the current table, or, if no label exists, to the name appearing after **from** in **load** and **select** statements.

Qualification is always turned off by default at the beginning of script execution. Qualification of a field name can be turned on at any time using a **qualify** statement. Qualification can be turned off at any time using an **unqualify** statement (see further on page 341).

The syntax is:

```
qualify ListOfFields
```

where

ListOfFields is one or more field names separated by commas.

The wildcard characters * and ? are allowed in the field names. It is also possible to replace *ListOfFields* by the character * to turn qualification on or off for all field names. Quoting of field names is sometimes necessary when using the wildcards as part of field names.

Note The **qualify** statement should not be used in conjunction with partial reload!

Examples:

```
Qualify B;  
Load A,B from x.csv;  
Load A,B from y.csv;
```

The two tables *x.csv* and *y.csv* will be joined only on *A*. Three fields will be found: *A*, *x.B*, *y.B*.

```
Qualify Comment;  
turns qualification on for all fields named Comment.
```

```
Qualify *;  
turns qualification on for all field names.
```

```
Qualify "ID";  
turns qualification on for all field names ending with ID.
```

```
Qualify "NR*",A;  
turns qualification on for A and all field names starting with NR.
```

```
Qualify F??;  
turns qualification on for three character field names starting with F.
```

In an unknown database, you may want to start by associating only one or a few fields:

```

Qualify *;
Unqualify TransID;
Select * from tab1;
Select * from tab2;
Select * from tab3;

```

Only *TransID* will be used for associations between the tables.

Rem

The **rem** statement is used to insert remarks in the script. Everything between the **rem** and the semicolon is considered to be a comment.

The syntax is:

```
rem string
```

where

string is an arbitrary text. (Often the text is a script statement that temporarily should not be used.)

Comments can also be inserted at any point in the script, except inside a quotation, if enclosed by `/*` and `*/`.

Furthermore, two double slashes `//` will cause the rest of the line in the script to be treated as a comment. Exception: `//`: can be part of a URL.

Note If you wish an external script specified by the **include** variable to be treated as a comment, you need to enclose it by `/*` and `*/` if the script contains several statements. Neither **rem** nor `//` can thus be used in this case.

Examples:

```

Rem ** This is a comment **;
Load a,b /* This is a comment */ from c.csv;
Load a,b from c.csv; // This is a comment
/* $(Include=abc.qvs); */

```

Rename Field

Renames one or more existing QlikView field(s).

The syntax is:

```
rename field[s] (using mapname | oldname to newname
{, oldname to newname } )
```

where

mapname is the name of a previously loaded mapping table containing one or more pairs of old and new field names

oldname is the old field name and
newname is the new field name.

Note Both **rename field** and **rename fields** are allowed forms with no difference in effect.

Examples:

```
Rename field XAZ0007 to Sales;  
FieldMap:  
Mapping select oldnames, newnames from datadict;  
Rename fields using FieldMap;
```

Rename Table

Renames one or more existing QlikView input table(s).

The syntax is:

```
rename table[s] (using mapname | oldname to newname  
{, oldname to newname } )
```

where

mapname is the name of a previously loaded mapping table containing one or more pairs of old and new table names

oldname is the old table name and

newname is the new table name.

Note Both **rename table** and **rename tables** are allowed forms with no difference in effect.

Examples:

```
Tab1:  
Select * from Trans;  
Rename table Tab1 to Xyz;  
  
TabMap:  
Mapping load oldnames, newnames from tabnames.csv;  
Rename tables using TabMap;
```

Replace

The **replace** prefix can be added to any **load**, **select** or **map..using** statement in the script. The **replace load** / **replace select** statement has the effect of dropping the entire QlikView table - whose table name is generated by the **replace load** / **replace select** statement - and replacing it with a new table

containing the result of the **replace load** / **replace select** statement. The effect is the same during partial and full reload. The **replace map..using** statement causes the mapping to take place also during partial reload.

The syntax is:

```
replace [ only ] ( loadstatement | selectstatement |
map..usingstatement )
```

where

only is an optional qualifier, which denotes that the statement should be disregarded during normal (non-partial) reloads.

Examples:

```
Tab1:
Replace Load * from File1.csv;
```

During both normal and partial reload, the QlikView table *Tab1* is first dropped and then new data is loaded from *File1.csv* and stored in *Tab1*.

```
Tab1:
Replace Only Load * from File1.csv;
```

During normal reload, this statement is disregarded.

During partial reload, any QlikView table previously named *Tab1* is first dropped and then new data is loaded from *File1.csv* and stored in *Tab1*.

```
Tab1:
Load a,b,c from File1.csv;
Replace Load a,b,c from File2.csv;
```

During normal reload, the file *File1.csv* is first read into the QlikView table *Tab1*, but then immediately dropped and replaced with new data from *File2.csv*. All data from *File1.csv* is lost.

During partial reload, the entire QlikView table *Tab1* is first dropped and then replaced with new data loaded from *File2.csv*.

```
Tab1:
Load a,b,c from File1.csv;
Replace only Load a,b,c from File2.csv;
```

During normal reload, data is loaded from *File1.csv* and stored in the QlikView table *Tab1*. *File2.csv* is disregarded.

During partial reload, the entire QlikView table *Tab1* is first dropped and then new data is loaded only from *File2.csv* and stored in *Tab1*. All data from *File1.csv* is lost.

See also the **add** keyword (page 284), which can be used to append data to tables during partial reload.

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Right

The **join** and **keep** prefixes in the QlikView script language can be preceded by the prefix **right**.

If used before **join**, it specifies that the join between the two tables should be a right join. The resulting table only contains combinations between the two tables with a full data set from the second table.

If used before **keep**, it specifies that the first table should be reduced to its common intersection with the second table before being stored in QlikView. See further on page 469.

The syntax is:

```
right ( join | keep ) [(tablename)] ( loadstatement | selectstatement )
```

Examples:

```
QVTable:  
Select * from Table1;  
right join select * from Table2;
```

```
QVTab1:  
Select * from Table1;  
QVTab2:  
right keep select * from Table2;
```

```
tab1:  
Load * from file1.csv;  
tab2:  
load * from file2.csv;  
.. .. ..  
right keep (tab1) load * from file3.csv;
```

The **outer** (page 323), **inner** (page 303), and **left** (page 307) prefixes can be used in similar ways.

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Sample

The **sample** prefix is used to load a random sample of records from the input table.

The syntax is:

```
sample p ( loadstatement | selectstatement )
```

where

p is an arbitrary expression which evaluates to a number larger than 0 and lower or equal to 1. The number indicates the probability for a given record to be read.

Examples:

```
Sample 0.15 Select * from Longtable;
Sample 0.15 Load * from Longtab.csv;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Section

With the **section** statement, it is possible to define whether the subsequent **load** and **select** statements should be considered as data or as a definition of the access rights.

The syntax is:

```
section ( access | application )
```

Examples:

```
Section access;
Section application;
```

If nothing is specified, **section application** is assumed. The section definition is valid until a new **section** statement is made.

Note All fields defined in the **section access** must be written in uppercase in the script.

Select

The selection of fields from an ODBC data source is made through standard SQL **select** statements. However, the extent to which the **select** statements are accepted depends largely on the ODBC driver used. Below a brief syntax description:

```
select [ all | distinct | distinctrow | top n [ percent ] ] *fieldlist
      from tablelist
      [ where criterion ]
      [ group by fieldlist [ having criterion ] ]
      [ order by fieldlist [ asc | desc ] ]
      [ ( inner | left | right | full ) join tablename on fieldref = fieldref
    ]
```

where

distinct is a predicate used if duplicate combinations of values in the selected fields only should be counted once.

distinctrow is a predicate used if duplicate records in the source table only should be counted once.

**fieldlist::= (* | field) { , field }*

A list of the fields to be selected. The "*" is used to select all the fields in the table.

fieldlist::= field { , field }

A list of one or more fields, separated by commas.

*field::= (fieldref | expression) [**as** aliasname]*

The expression can e.g. be a numeric or string function based on one or several other fields. Some of the operators and functions usually accepted are: +, -, *, /, & (string concatenation), sum(fieldname), count(fieldname), avg(fieldname) (average), month(fieldname), etc. See the documentation of your ODBC driver for more information.

fieldref::= [tablename .] fieldname

The tablename and the fieldname are text strings identical to what they imply. They must be enclosed by straight double quotation marks if they contain e.g. spaces.

The **as** clause is used to assign a new name to the field.

tablelist::= table { , table }

The list of tables that the fields are to be selected from.

*table::= tablename [[**as**] aliasname]*

The tablename may or may not be put within quotes.

where is a clause used to state whether a record should be included in the selection or not.

criterion is a logical expression that can sometimes be very complex. Some of the operators accepted are: numeric operators and functions, =, <> or # (not equal), >, >=, <, <=, and, or, not, exists, some, all, in and also new **select** statements. See the documentation of your ODBC driver for more information.

group by is a clause used to aggregate (group) several records into one. Within one group, for a certain field, all the records must either have the same value, or the field can only be used from within an

expression, e.g. as a sum or an average. The expression based on one or several fields is defined in the expression of the field symbol. **having** is a clause used to qualify groups in a similar manner as the **where** clause is used to qualify records.

order by is a clause used to state the sort order of the resulting table of the **select** statement.

join is a qualifier stating if several tables are to be joined together into one.

Field names and table names must be put within quotes if they contain spaces or letters from the national character sets. When the script is automatically generated by QlikView, the quotation mark used is the one preferred by the ODBC driver specified in the data source definition of the data source in the **connect** statement.

Furthermore, several **select** statements can sometimes be concatenated into one through the use of a **union** operator:

selectstatement **union** *selectstatement*

The **select** statement is interpreted by the ODBC driver, so deviations from the general SQL syntax may occur depending on the capabilities of the ODBC drivers, e.g.

- **as** is sometimes not allowed, i.e. *aliasname* must follow immediately after *fieldname*.
- **as** is sometimes compulsory if an *aliasname* is used.
- **distinct**, **as**, **where**, **group by**, **order by**, or **union** are sometimes not supported.
- the ODBC driver sometimes does not accept all the different quotation marks listed above.

Note This is not a complete description of the SQL **select** statement! E.g., **select** statements can be nested, several joins can be made in one **select** statement, the number of functions allowed in expressions is sometimes very large, etc.

Examples:

```
Select * from `Categories`;
Select `Category ID`, `Category Name` from
`Categories`;
Select `Order ID`, `Product ID`,
`Unit Price` * Quantity * (1-Discount) as NetSales
from `Order Details`;
Select `Order Details`.`Order ID`,
Sum(`Order Details`.`Unit Price` * `Order
Details`.Quantity) as `Result`
from `Order Details`, Orders
where Orders.`Order ID` = `Order Details`.`Order ID`
group by `Order Details`.`Order ID`;
```

Semantic

Tables containing relations between objects can be loaded using the **semantic** prefix. See further page 483.

The syntax is:

```
semantic ( loadstatement | selectstatement )
```

Examples:

```
Semantic Load * from abc.csv;
Semantic Select Object1, Relation, Object2,
InverseRelation from table1;
```

The **load** statement syntax is described on page 309. The **select** statement syntax is described on page 329.

Tables loaded using a **semantic** prefix cannot be concatenated.

Set

The **set** statement is used to define script variables. These can be used to substitute strings, paths, drives, etc. See further page 343. If the expression on the right side of the '=' should be evaluated before it is assigned to the macro variable, use the **let** statement (described on page 308) instead.

The syntax is:

```
set variablename =string
```

Examples:

```
Set FileToUse=Data1.csv;
Set Constant="My string";
Set BudgetYear=1997;
Set YearRel = "RecNo()";
Set Year = "$ (YearRel) + 1997";
```

Examples showing the difference between set and let:

```
Set X=3+4;
Let Y=3+4;
Let Z=$ (Y) +1;
```

\$ (X) will be evaluated as '3+4', whereas **\$ (Y)** will be evaluated as '7'.
Finally, **\$ (Z)** will be evaluated as '8'.

Sleep

Pauses script execution *n* milliseconds, where *n* is a positive integer no larger than 3600000 (equals 1 hour wait). The value may be an expression. The syntax is:

```
sleep n
```

Examples:

```
sleep 10000;
sleep t*1000;
```

SQL

The SQL statement allows you to send an arbitrary SQL command through an ODBC or OLE DB connection.

The syntax is:

```
SQL command
```

where

command is a valid SQL statement.

Example:

```
SQL Leave;
```

Sending SQL statements that update the database will return an error, unless the ODBC or OLE DB connection has been opened with the **mode is write** qualifier.

The syntax:

```
SQL select * from tabl;
```

is allowed and will in future documentation be the preferred syntax for **select** (for consistency reasons). The SQL prefix will however remain optional for **select** statements.

SQLColumns

The **sqlcolumns** statement returns a set of fields describing the columns of an ODBC data source to which a connection has been made. These fields can

be combined with the fields generated by the **sqltables** and **sqltypes** commands in order to give a good overview of a database.

The standard fields are:

TABLE_QUALIFIER
TABLE_OWNER
TABLE_NAME
COLUMN_NAME
DATA_TYPE
TYPE_NAME
PRECISION
LENGTH
SCALE
RADIX
NULLABLE
REMARKS

For a detailed description of these fields, see an ODBC reference handbook.

Note Some ODBC drivers may not support this command.

Some ODBC drivers may produce additional fields.

The syntax is:

sqlcolumns

Example:

```
CONNECT TO 'MS Access 7.0
Database;DBQ=C:\Course3\DataSrc\QWT.mdb';
sqlcolumns;
```

SQLTables

The **sqltables** statement returns a set of fields describing the tables of an ODBC data source to which a connection has been made. These fields can be combined with the fields generated by the **sqlcolumns** and **sqltypes** commands in order to give a good overview of a database.

The standard fields are:

TABLE_QUALIFIER
TABLE_OWNER
TABLE_NAME
TABLE_TYPE
REMARKS

For a detailed description of these fields, see an ODBC reference handbook.

Note Some ODBC drivers may not support this command.

Some ODBC drivers may produce additional fields.

The syntax is:

sqltables

Example:

```
CONNECT TO 'MS Access 7.0
Database;DBQ=C:\Course3\DataSrc\QWT.mdb';
sqltables;
```

SQLTypes

The **sqltypes** statement returns a set of fields describing the types of an ODBC data source to which a connection has been made. These fields can be combined with the fields generated by the **sqlcolumns** and **sqltables** commands in order to give a good overview of a database.

The standard fields are:

```
TYPE_NAME
DATA_TYPE
PRECISION
LITERAL_PREFIX
LITERAL_SUFFIX
CREATE_PARAMS
NULLABLE
CASE_SENSITIVE
SEARCHABLE
UNSIGNED_ATTRIBUTE
MONEY
AUTO_INCREMENT
LOCAL_TYPE_NAME
MINIMUM_SCALE
MAXIMUM_SCALE
```

For a detailed description of these fields, see an ODBC reference handbook.

Note Some ODBC drivers may not support this command. Some ODBC drivers may produce additional fields.

The syntax is:

sqltypes

Example:

```
CONNECT TO 'MS Access 7.0
Database;DBQ=C:\Course3\DataSrc\QWT.mdb';
sqltypes;
```

Star

The string used to represent the set of all the values of a field in the database can be set through the **star** statement. It affects the subsequent **load** and **select** statements.

The syntax is:

```
star is [ string ]
```

where

string is an arbitrary text. Note that the string must be enclosed by quotation marks if it contains blanks.

Examples:

```
Star is *;
Star is %;
Star is;
```

If nothing is specified, **star is;** is assumed, i.e. there is no star symbol available unless explicitly specified. This definition is valid until a new **star** statement is made. See further page 491.

Store

A QVD or a CSV file can be created by a **store** statement in the script. The statement will create an explicitly named QVD or CSV file. The statement can only export fields from one logical table. The text values are exported to the CSV file in UTF-8 format. A delimiter can be specified, see **load** examples on page 312. The **store** statement to a CSV file does not support BIFF export.

The syntax is:

```
store [*fieldlist from] table into filename[format-spec];
```

where

```
*fieldlist::= ( * | field { , field } )
```

A list of the fields to be selected. The " *" is used to select all the fields in the table.

```
field::= fieldname [ as aliasname ]
```

```
format-spec::=((txt/qvd)),
```


The format specification consists of the text **txt** for text files, or the text **qvd** for qvd files. If the format specification is omitted, **qvd** is assumed.

fieldname is a text that is identical to a field name in the table. Note that the field name must be enclosed by straight double quotation marks or square brackets if it contains e.g. spaces.

aliasname is an alternate name for the field to be used in the resulting QVD or CSV file. Note that the name must be enclosed by straight double quotation marks or square brackets if it contains e.g. spaces.

table is a script labeled, already loaded table to be used as source for data.

filename is the name of the target file. The file name is interpreted similar to names in LOAD statements, i.e. the DIRECTORY statements apply.

Examples:

```
store mytable into xyz.qvd;
store * from mytable into xyz.qvd (qvd);
store Name, RegNo from mytable into xyz.qvd;
store Name as a, RegNo as b FROM mytable into xyz.qvd;
store mytable into myfile.txt (txt);
store * from mytable into myfile.txt (txt);
```

The two first statements have identical function.

Sub..End Sub

The **sub** control statement defines a subroutine, which can be called upon from a **call** statement.

The syntax is:

```
sub name [ ( paramlist ) ]
      statements
end sub
```

where

name is the name of the subroutine.

paramlist is a comma separated list of variable names for the formal parameters of the subroutine.

statements is any group of one or more QlikView script statements.

Parameters are copied into the subroutine and, if the corresponding parameter in the calling **call** statement is a variable name, copied back out again upon exit from the subroutine.

If a subroutine has more formal parameters than actual parameters passed by a **call** statement, the extra parameters will be initialized to NULL and can be used as local variables within the subroutine.

Since the **sub** statement is a control statement, each of its two clauses (**sub** and **end sub**) must not cross a line boundary. They may be terminated by semicolon or end-of-line.

Examples:

```
// example 1
sub INCR (I,J)
    I = I + 1
    exit sub when I < 10
    J = J + 1
end sub
call INCR (X,Y)

// example 2 - parameter transfer
sub ParTrans (A,B,C)
    A=A+1
    B=B+1
    C=C+1
end sub

A=1
X=1
C=1
call ParTrans (A,(X+1)*2)
// A=2 (copied back from sub)
// X=1 (expression not copied back)
// C=1 (C in sub is local)
// B=null (local in sub)
```

Switch..Case..Default..End Switch

The **switch** control statement creates a fork where one of several groups of statements is executed, depending on the value of an *expression*.

The syntax is:

```
switch expression
{ case valuelist
  [ statements ] }
[ default
```

statements]

end switch

where

expression is an arbitrary expression.

valuelist is a comma separated list of values with which the value of *expression* will be compared. Execution of the script will continue with the statements in the first **case** clause encountered with a value in *valuelist* equal to the value of *expression*. Each value in *valuelist* may be an arbitrary expression. If no match is found in any **case** clause, the statements under the **default** clause, if it exists, will be executed.

statements is any group of one or more QlikView script statements.

Since the **switch** statement is a control statement, each of its four possible clauses (**switch**, **case**, **default** and **end switch**) must not cross a line boundary. They may be terminated by semicolon or end-of-line.

Examples:

```
switch I
  case 1
    load '$(I): CASE 1' as case autogenerate 1;
  case 2
    load '$(I): CASE 2' as case autogenerate 1;
  default
    load '$(I): DEFAULT' as case autogenerate 1;
end switch
```

Trace

Writes string to the script progress dialog window and to the script log file, if such is used. The syntax is:

trace *string*

Example:

```
trace Main table loaded;
trace $(MyMessage);
```

Unless

The **unless** prefix and suffix is used to create a conditional clause, which determines whether a statement or an **exit** clause should be executed or not. It can be seen as a compact alternative to a full **if..end if** statement.

The syntax is:

(**unless** *condition statement* | *exitstatement* **unless** *condition*)

where

condition is a logical expression evaluating to true or false.

statement is any QlikView script statement except control statements.

exitstatement is an **exit for**, **exit do** or **exit sub** clause or an **exit script** statement.

The *statement* or *exitstatement* will only be executed if *condition* is evaluated to false.

The **unless** prefix may be used on statements which already have one or more other prefixes, including additional **when** and **unless** prefixes.

Examples:

```
exit script unless A=1
unless A=1 load * from myfile.csv;
unless A=1 when B=2 drop table Tab1;
```

Unmap

The **unmap** statement disables mapping of specified fields from the current point in the script.

The syntax is:

```
unmap fieldlist
```

where

```
fieldlist := *| ( field { , field } )
```

Using * as *fieldlist* indicates all fields.

field is the name of a field that should no longer be mapped from this point in the script. The wildcard characters * and ? are allowed in the field names. Quoting of field names may be necessary when using the wildcards.

Examples:

```
UnMap Country;
disables mapping of the field Country.
```

```
UnMap A, B, C;
disables mapping of the fields A, B and C.
```

```
UnMap *;
disables mapping of all fields.
```

```
UnMap "A*", "?B?";
disables mapping of fields starting with A and three character fields with B as second character.
```

See also “Mapping” on page 318 and “Map..Using” on page 319, as well as page 488.

Unqualify

The **unqualify** statement turns off qualification of field names previously set by a **qualify** statement. See further on page 341.

The syntax is:

```
unqualify ListOfFields
```

where

ListOfFields is one or more field names separated by commas.

The wildcard characters * and ? are allowed in the field names. It is also possible to replace *ListOfFields* by the character * to turn qualification on or off for all field names. Quoting of field names is sometimes necessary when using the wildcards as part of field names.

Examples:

```
Unqualify Comment;
```

turns qualification off for all fields named *Comment*.

```
Unqualify *;
```

turns qualification off for all field names.

```
Unqualify "*ID";
```

turns qualification off for all field names ending with *ID*.

```
Unqualify "NR*", A;
```

turns qualification off for *A* and all field names starting with *NR*.

```
Unqualify F??;
```

turns qualification off for three character field names starting with *F*.

In an unknown database, you may want to start by associating only one or a few fields:

```
Qualify *;  
Unqualify TransID;  
Select * from tab1;  
Select * from tab2;  
Select * from tab3;
```

Only *TransID* will be used for associations between the tables.

When

The **when** prefix and suffix is used to create a conditional clause, which determines whether a statement or an **exit** clause should be executed or not. It can be seen as a compact alternative to a full **if..end if** statement.

The syntax is:

```
( when condition statement | exitstatement when condition )
```

where

condition is a logical expression evaluating to true or false.

statement is any QlikView script statement except control statements.

exitstatement is an **exit for**, **exit do** or **exit sub** clause or an **exit script** statement.

The *statement* or *exitstatement* will only be executed if *condition* is evaluated to true.

The **when** prefix may be used on statements which already have one or more other prefixes, including additional **when** and **unless** prefixes.

Examples:

```
exit script when A=1
when A=1 load * from myfile.csv;
when A=1 unless B=2 drop table Tab1;
```

21.3 Table Names

Naming of Tables

QlikView input tables are named as they are stored in the QlikView internal database. The table names can be used e.g. for **load...resident**, **peek** etc. and can be seen in the *\$Table* system field in the layout.

Tables are named according to the following rules:

- 1 If a label immediately precedes a **load** or **select** statement the label will be used as table name. The label must be followed by a colon (see below under *Table labels*).
- 2 If no label is given, the filename or table name immediately following the keyword **from** in the **load** or **select** statement is used. A maximum of 32 characters is used. For filenames the extension is skipped.

- 3 Tables loaded inline will be named `INLINExx`, where `xx` is a number. The first inline table will be given the name `INLINE01`.
- 4 Automatically generated tables will be named `AUTOGENERATExx`, where `x` is a number. The first autogenerated table will be given the name `AUTOGENERATE01`.
- 5 If a table name generated according to the rules above should be in conflict with a previous table name, the name will be extended with `-x`, where `x` is a number. The number is increased until no conflict remains. For example, three tables could be named *Budget*, *Budget-1* and *Budget-2*.

There are three separate domains for table names: **section access**, **section application** and mapping tables. Table names generated in **section access** and **section application** will be treated separately. If a table name referenced is not found within the section, QlikView will search the other section as well. Mapping tables are treated separately and have no connection whatsoever to the other two domains of table names.

Table Labels

An input table can be labeled for later reference by a **load** statement with a **resident** clause or with expressions containing the **peek** function. The label, which can be an arbitrary string of numbers and characters, should precede the first **load** or **select** statement that creates the input table. The label must end with a colon.

Examples:

```
Table1:
Load a,b from c.csv;
...
Load x,y from d.csv where x=peek("a",y,Table1);
Transactions:

Select * from Transtable;
...
Load Month, sum(Sales) resident Transactions group by
Month;
```

21.4 Script Variables

Script variables are entities that can be assigned any text or numeric value. When used, the variable is substituted by its value. Variables can be used in the script for macro expansion and in various control statements.

Some special system variables will be set by QlikView at the start of the script execution regardless of their previous values.

When defining a script variable, the following syntax is used

set *variablename* = *string*

or the syntax

let *variable* = *expression*

The **Set** command assigns the variable the text to the right of the equal sign, whereas the **Let** command evaluates the expression.

Variables are case sensitive.

Example:

```
set HidePrefix = $ ; // the variable will get the character '$' as value.
```

```
let vToday = Num(Today()); // returns the date serial number of today.
```

Special Variables

There are some variable names that have special meanings in QlikView:

Floppy

Returns the drive letter of the first floppy drive found, normally **a:**. A system defined variable.

CD

Returns the drive letter of the first CD-ROM drive found. If no CD-ROM is found, then **c:** is returned. A system defined variable.

QvPath

Returns the browse string to the QlikView executable. A system-defined variable.

QvRoot

Returns the root directory of the QlikView executable. A system-defined variable.

QvWorkPath

Returns the browse string to the current QlikView document. A system-defined variable.

QvWorkRoot

Returns the root directory of the current QlikView document. A system-defined variable.

WinPath

Returns the browse string to Windows. A system-defined variable.

WinRoot

Returns the root directory of Windows. A system-defined variable.

Include

This variable specifies a file that contains text that should be included in the script. The entire script can thus be put in a file. It is called through

```
$(Include=filename)
```

Example:

```
$(Include=abc.txt)
```

I.e. the construction **set Include=filename** is not applicable.

HidePrefix

All field names beginning with this text string will be hidden in the same manner as the system fields. A user-defined variable.

Example:

```
set HidePrefix='_';
```

If this statement is used, the field names beginning with an underscore will not be shown in the field name lists when the system fields are hidden.

HideSuffix

All field names ending with this text string will be hidden in the same manner as the system fields. A user-defined variable.

Example:

```
set HideSuffix='%';
```

If this statement is used, the field names ending with a percentage sign will not be shown in the field name lists when the system fields are hidden.

StripComments

If this variable is set to 0, stripping of `/*..*/` and `//` comments in the script will be inhibited. This may be useful with certain database drivers which use `/*..*/` comments to define hints in select statements. It is recommended to reset the variable to 1 after the statement(s) where it is needed in order to avoid errors in script execution. If the variable is not defined in a document, stripping of comments will always take place.

Example:

```
set StripComments=0;
```

Verbatim

Normally all field values are stripped of preceding and succeeding blank characters (ASCII 32) before being loaded into the QlikView database. By setting this variable to 1 this stripping is suspended.

Example:

```
set Verbatim=1;
```

OpenUrlTimeout

This variable defines the time-out in seconds that QlikView should respect when getting data from URL sources (e.g. HTML pages). If omitted the time-out is about 20 minutes.

Example:

```
set OpenUrlTimeout=10
```

Value Handling Variables

NullDisplay

The defined symbol will substitute all NULL values from ODBC on the lowest level of data. See “NULL Value Handling in QlikView” on page 492.

Example:

```
set NullDisplay='<NULL>';
```

NullInterpret

The defined symbol will when it occurs in a text file or an **inline** statement be interpreted as NULL. See “NULL Value Handling in QlikView” on page 492.

Example:

```
set NullInterpret='';
```

NullValue

The defined symbol will substitute all NULL values that are considered to be unknown rather than undefined. The only NULL values affected are those included in the **nullasvalue** statement (see page 321)

Example:

```
set NullValue='<NULL>';
```

OtherSymbol

Defines a symbol to be treated as 'all other values' before a **load/**
select statement. See page 491.

Example:

```
set OtherSymbol='+';
```

Number Interpretation Variables

The following variables will be automatically generated according to the current operating system settings when a new document is created. They are used to substitute the operating system defaults for certain number formatting settings at the time of the script execution. They may be deleted, edited or duplicated freely. These variables can make a QlikView document portable to computers with other default settings.

ThousandSep

The thousand separator defined replaces the digit grouping symbol of the operating system (Control Panel).

Example:

```
Set ThousandSep=',';
```

DecimalSep

The decimal separator defined replaces the decimal symbol of the operating system (Control Panel).

Example:

```
Set DecimalSep='.';
```

MoneyThousandSep

The thousand separator defined replaces the digit grouping symbol for currency of the operating system (Control Panel).

Example:

```
Set MoneyThousandSep=',';
```

MoneyDecimalSep

The decimal separator defined replaces the decimal symbol for currency of the operating system (Control Panel).

Example:

```
Set MoneyDecimalSep='.';
```

MoneyFormat

The symbol defined replaces the currency symbol of the operating system (Control Panel).

Example:

```
Set MoneyFormat='#,##0.00; (#,##0.00)';
```

TimeFormat

The format defined replaces the time format of the operating system (Control Panel).

Example:

```
Set TimeFormat='hh:mm:ss';
```

DateFormat

The format defined replaces the date format of the operating system (Control Panel).

Example:

```
Set DateFormat='M/D/YY';
```

TimestampFormat

The format defined replaces the date and time formats of the operating system (Control Panel).

Example:

```
Set TimestampFormat='M/D/YY hh:mm:ss[.fff]';
```

MonthNames

The month names defined replace standard short month names of the operating system (Regional Settings).

Example:

```
Set MonthNames='Jan;Feb;Mar;Apr;...';
```

DayNames

The day names defined replace standard short weekday names of the operating system (Regional Settings).

Example:

```
Set DayNames='Mon;Tue;Wed;Thu;Fri...';
```

LongMonthNames

The month names defined replace standard month names of the operating system (Regional Settings).

Example:

```
Set LongMonthNames='January;February;...';
```

LongDayNames

The day names defined replace standard weekday names of the operating system (Regional Settings).

Example:

```
Set LongDayNames= 'Monday;Tuesday;...';
```

Error Variables

There are four special variables which can be used for error handling in the script:

ErrorMode

Determines the action to be taken by QlikView when an error is encountered during script execution. By default (ErrorMode=1) the script execution will halt and the user will be prompted for action (non-batch mode). By setting ErrorMode=0 QlikView will simply ignore the failure and continue script execution at the next script statement. By setting ErrorMode=2 QlikView will trigger a "Execution of script failed..." error message immediately on failure, without prompting the user for action beforehand.

Example:

```
set ErrorMode=0;
```

ScriptError

The error code of the last executed script statement. This variable will be reset to 0 after each successfully executed script statement. If an error occurs it will be set to an internal QlikView error code. The error code is a dual value with a numeric and a text value. Error codes are as follows:

- | | |
|----|-----------------------|
| 1 | No Error |
| 2 | General Error |
| 3 | Syntax Error |
| 4 | General ODBC Error |
| 5 | General OLEDB Error |
| 6 | General XML Error |
| 7 | General HTML Error |
| 8 | File Not Found |
| 9 | Database Not Found |
| 10 | Table Not Found |
| 11 | Field Not Found |
| 12 | File Has Wrong Format |

Examples:

```
set ErrorMode=0;
load * from abc.csv;
if ScriptError=8 then
exit script;
//no file;
end if
```

```
set ErrorMode=0;
load * from abc.csv;
if '$(ScriptError)'='File Not Found' then
exit script;
end if
```

ScriptErrorDetails

Returns a more detailed error description for some of the error codes above. Most importantly this variable will contain the error message returned by ODBC and OLEDB drivers for error codes 3 and 4.

ScriptErrorCount

Returns the total number of statements that have caused errors during the current script execution. This variable is always reset to 0 at the start of script execution.

Example:

```
Set ErrorMode=0;
Load * from abc.csv;
if ScriptErrorCount >= 1 then
exit script;
end if
```

ScriptErrorList

This variable will contain a concatenated list of all script errors that have occurred during the last script execution. Each error is separated by a line feed.

The values of all variables above will remain after script execution. The value of **ScriptError**, **ScriptErrorDetailed**, **ScriptErrorCount** and **ScriptErrorList** for error handling inside the script is of course dependent on the use of `ErrorMode=0`.

21.5 Dollar-Sign Expansions

Dollar-sign expansions are definitions of text replacements used in the script or in expressions. This process is known as expansion - even if the new text is shorter. The

replacement is made just before the script statement or the expression is evaluated. Technically it is a macro expansion.

A macro expansion always begins with '\$(' and ends with ')' and the content between brackets defines how the text replacement will be done. To avoid confusion with script macros we will henceforth refer to macro expansions as dollar-sign expansions.

Note Macro expansion is unrelated to script macros (VB or Java script defined in the script module).

Dollar-Sign Expansion using a variable

When using a variable for text replacement in the script or in an expression, the syntax

`$(variablename)`

is used. `$(variablename)` expands to the value in *variablename*. If this is a number or a date, the string will format according to active regional settings. If *variablename* does not exist the expansion will be the empty string.

For numeric variable expansions the syntax

`$(#variablename)`

is used. `$(#variablename)` always yields a legal decimal-point number reflection of the numeric value of *variablename*, possibly with exponential notation (for very large/small numbers). If *variablename* does not exist or does not contain a numeric value, it will be expanded to 0 instead.

After execution of the following script:

```
SET DecimalSep=',';
LET X = 7/2;
```

`$(X)` will expand to 3,5 while `$(#X)` will expand to 3.5.

Examples:

```
set MyPath=C:\MyDocs\Files\
...
load * from $(MyPath)abc.csv;
set CurrentYear=1992;
...
select * from table1 where Year=$(CurrentYear);
```

Dollar-Sign Expansion with Parameters

Parameters can be used in variable expansions. The variable must then contain formal parameters, such as \$1, \$2, \$3 etc. When expanding the variable, the parameters should be given in a comma separated list:

Examples:

```
set MUL=' $1*$2';  
set X=$(MUL(3,7)); // returns '3*7' in X  
let X=$(MUL(3,7)); // returns 21 in X
```

If the number of formal parameters exceeds the number of actual parameters only the formal parameters corresponding to actual parameters will be expanded. If the number of actual parameters exceeds the number of formal parameters the superfluous actual parameters will be ignored.

Examples:

```
set MUL=' $1*$2';  
set X=$(MUL); // returns '$1*$2' in X  
set X=$(MUL(10)); // returns '10*$2' in X  
let X=$(MUL(5,7,8)); // returns 35 in X
```

The parameter **\$0** returns the number of parameters actually passed by a call.

Example:

```
set MUL=' $1*$2 $0 par';  
set X=$(MUL(3,7)); // returns '3*7 2 par' in X
```

Dollar-Sign Expansion with an Expression

Expressions can be used in dollar-sign expansions. The content between the brackets must then start with an equal sign:

\$(=expression)

The expression will be evaluated and the value will be used in the expansion.

Example:

```
$(=Year(Today())); // returns e.g. '2008'  
$(=Only(Year)-1); // returns the year before the selected one
```

Dollar-Sign Expansion of File – Include statement

File inclusions are made using dollar-sign expansions. The syntax is then:

\$(include=filename)

The above text will be replaced by the content of the file specified after the equal sign. This feature is very useful if you want to store scripts or parts of scripts in text files.

Example:

```
$ (include=C:\Documents\MyScript.qvs) ;
```

21.6 Quotation Marks

If quotation marks are possible or necessary within a statement, it is mentioned below. In a **load** statement the following symbols should be used as quotation marks:

Field names: [], " ", `` or ´ ´

String literals: ´ ´

For a **select** statement interpreted by the ODBC driver it may be slightly different, though. Usually you should use the straight double quotation marks (alt+0034) for names and the straight single quotation marks (alt+0039) for literals, and thus never the oblique: ´ ´ or ``. However, some ODBC drivers not only accept the oblique quotation marks, but also prefer them. In such a case the generated **select** statements contain oblique quotation marks.

Example: Difference between Names and Literals

The difference between names and literals becomes clearer if you compare the following expressions:

```
'Sweden' as Country
```

as a part of the field list in a **load** or **select** statement will load the text string *Sweden* as field value into the QlikView field *Country*, whereas

```
"land" as Country
```

as a part of the field list in a **load** or **select** statement will load the content of the database field or table column named *land* as field values into the QlikView field *Country*. I.e. *land* will be treated as a field reference.

Example: Difference between Numbers and String Literals

The difference between names and string literals becomes clearer if you compare the following expressions:

```
'12/31/96'
```

as a part of an expression will be interpreted as the text string *12/31/96*, whereas

```
12/31/96
```

as a part of an expression will be interpreted numerically as the date *Dec 12, 1996*, and assigned the corresponding numeric value, provided that the system date format is MM/DD/YY.

Example: MS Access

Microsoft Access ODBC Driver 3.4 (included in MS Access 7.0) accepts the following quotes when analyzing the select statement.

Field names and table names: [], " " or ``

String literals: ''

Not accepted: ^^

22 SCRIPT EXPRESSIONS

Expressions can be used in both the **load** statement and the **select** statement. The syntax and functions described here apply to the **load** statement, and not to the **select** statement, since the latter is interpreted by the ODBC driver and not by QlikView. However, most ODBC drivers are often capable of interpreting a number of the functions described below.

All expressions in a QlikView script return a number and/or a string, whichever is appropriate. Logical functions and operators return 0 for false and -1 for true. Number to string conversions and vice versa are implicit. Logical operators and functions interpret 0 as false and all else as true.

22.1 Expression Syntax

The general syntax for an expression is:

```

expression ::= ( constant |
                  fieldref |
                  operator1 expression |
                  expression operator2 expression |
                  function |
                  ( expression ) )

```

where

constant is a string (a text, a date or a time) enclosed by single straight quotation marks, or a number. Constants are written with no thousand separator and with a decimal point as decimal separator.

fieldref is a field name of the loaded table. (See “Load” on page 309.)

operator1 is a unary operator (working on one expression, the one to the right).

operator2 is a binary operator (working on two expressions, one on each side).

function ::= *functionname* (*parameters*)

parameters ::= *expression* { , *expression* }

The number and types of parameters is not arbitrary. It depends on the function used.

Expressions and functions can thus be nested freely, and as long as the expression returns an interpretable value, QlikView will not give any error messages.

22.2 Operators

There are two types of operators in QlikView, unary operators that take only one operand, and binary operators that take two operands. Most operators are binary.

Numeric Operators

All numeric operators use the numeric values of the operands and return a numeric value as result.

+

Sign for positive number (unary operator) or arithmetic addition. The binary operation returns the sum of the two operands.

-

Sign for negative number (unary operator) or arithmetic subtraction. The unary operation returns the operand multiplied by -1, and the binary the difference between the two operands.

Arithmetic multiplication. The operation returns the product of the two operands.

/

Arithmetic division. The operation returns the ratio between the two operands.

String Operators

There are two string operators. One uses the string values of the operands and returns a string as result. The other one compares the operands and returns a boolean value to indicate match.

&

String concatenation. The operation returns a text string that consists of the two operand strings, one after another.

Example:

'abc' & 'xyz' returns 'abcxyz'

like

String comparison with wildcard characters. The operation returns a boolean true (-1) if the string before the operator is matched by the string after the operator. The second string may contain the wildcard

characters * (any number of arbitrary characters) or ? (one arbitrary character).

Examples:

'abc' like 'a*' returns true (-1)

'abcd' like 'a?c*' returns true (-1)

'abc' like 'a??bc' returns false (0)

Logical Operators

All logical operators interpret the operands logically and return true (-1) or false (0) as result.

not

Logical inverse. One of the few unary operators. The operation returns the logical inverse of the operand.

and

Logical and. The operation returns the logical and of the operands.

or

Logical or. The operation returns the logical or of the operands.

xor

Logical exclusive or. The operation returns the logical exclusive or of the operands. I.e. like logical or, but with the difference that the result is false if both operands are true.

Relational Operators

All relational operators compare the values of the operands and return true (-1) or false (0) as result. All relational operators are binary.

<

Less than. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

<=

Less than or equal. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

>

Greater than. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

>=

Greater than or equal. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

=

Equals. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

<>

Not equivalent to. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

follows

ASCII greater than. Returns true if the left-hand operand has a text representation which, in an ASCII comparison, comes after the text representation of the right-hand operand. Unlike the > operator no attempt is made to make a numeric interpretation of the argument values before the comparison.

Examples:

'23' **follows** '111' returns true

'23' > '111' returns false

precedes

ASCII less than. Returns true if the left-hand operand has a text representation which, in an ASCII comparison, comes before the text representation of the right-hand operand. Unlike the < operator no attempt is made to make a numeric interpretation of the argument values before the comparison.

Examples:

'11' **precedes** '2' returns true

'11' < '2' returns false

Bit operators

All bit operators convert the operands to signed integers (32 bit) and return the result in that way. All operations are performed bit by bit.

bitnot

Bit inverse. Unary operator. The operation returns the logical inverse of the operand performed bit by bit.

bitand

Bit and. The operation returns the logical and of the operands performed bit by bit.

bitor

Bit or. The operation returns the logical or of the operands performed bit by bit.

bitxor

Bit exclusive or. The operation returns the logical exclusive or of the operands performed bit by bit.

>>

Bit right shift. Unary operator. The operation returns the operand shifted one step to the right.

<<

Bit left shift. Unary operator. The operation returns the operand shifted one step to the left.

22.3 Aggregation Functions

The QlikView script language supports more than 110 standard functions. These have been divided into groups as follows:

Basic Aggregation Functions	page 359
String Aggregation Functions.....	page 361
Counter Aggregations Functions.....	page 362
Statistical Aggregation Functions.....	page 364
Financial Aggregation Functions	page 371
Statistical Test Functions	page 372
More functions are available in QlikView, see “Other Functions” on page 383.	

Basic Aggregation Functions

These functions can only be used in field lists for **load** statements with a **group by** clause.

sum([distinct] expression)

Returns the sum of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, sum(Sales) as SalesPerMonth
from abc.csv group by Month;
```

min(*expression* [, *rank*])

Returns the minimum numeric value of *expression* encountered over a number of records as defined by a **group by** clause. *Rank* defaults to 1 which correspond to the lowest value. By specifying *rank* as 2 the second lowest value will be returned. If *rank* is 3 the third lowest value will be returned and so on.

Examples:

```
Load Month, min(Sales) as
SmallestSalePerMonth
from abc.csv group by Month;

Load Month, min(Sales,2) as
SecondSmallestSalePerMonth
from abc.csv group by Month;
```

max(*expression* [, *rank*])

Returns the maximum numeric value of *expression* encountered over a number of records as defined by a **group by** clause. *Rank* defaults to 1 which correspond to the highest value. By specifying *rank* as 2 the second highest value will be returned. If *rank* is 3 the third highest value will be returned and so on.

Examples:

```
Load Month, max(Sales) as LargestSalePerMonth
from abc.csv group by Month;

Load Month, max(Sales,2) as
SecondLargestSalePerMonth
from abc.csv group by Month;
```

only(*expression*)

If *expression* over a number of records as defined by a **group by** clause contains only one value, that value is returned, else NULL is returned. **Only** can return numeric values as well as text values.

Examples:

```
Load Month, only(Price) as OnlyPriceSoldFor
from abc.csv group by Month;

Load Month, only(Salesman) as OnlyPersonWhoSold
from abc.csv group by Month;
```

mode(*expression*)

Returns the mode value, i.e. the most commonly occurring value, of *expression* over a number of records as defined by a **group by** clause. If more than one value is equally commonly occurring,

NULL is returned. **Mode** can return numeric values as well as text values.

Examples:

```
Load Month, mode(ErrorNumber) as
MostCommonErrorNumber
from abc.csv group by Month;

Load Month, mode(Product) as
ProductMostOftenSold
from abc.csv group by Month;
```

firstsortedvalue ([**distinct**] *expression* [, *sort-weight* [, *n*]]

returns the first value of *expression* sorted by corresponding *sort-weight* when *expression* is iterated over a number of records as defined by a **group by** clause. *Sort-weight* should return a numeric value where the lowest value will render the corresponding value of *expression* to be sorted first. By preceding the *sort-value* expression with a minus sign, the function will return the last value instead. If more than one value of expression share the same lowest *sort-order*, the function will return null. By stating an *n* larger than 1, you will get the *n*th value in order. If the word **distinct** occurs before the *expression*, all duplicates will be disregarded.

Example:

```
Load Customer,
firstsortedvalue(PurchasedArticle, OrderDate) as
FirstProductBought
from abc.csv group by Customer;
```

String Aggregation Functions

MinString(*expression*)

Returns the first text value in text sort order of *expression* over a number of records as defined by a **group by** clause. If no text value is found, NULL is returned.

Example:

```
Load City, MinString(Name) as FirstName
from abc.csv group by City;
```

MaxString(*expression*)

Returns the last text value in text sort order of *expression* over a number of records as defined by a **group by** clause. If no text value is found, NULL is returned.

Example:

```
Load City, MaxString(Name) as LastName
from abc.csv group by City;
```

FirstValue(*expression*)

Returns the first value in load order of *expression* over a number of records as defined by a **group by** clause. If no text value is found, NULL is returned.

Example:

```
Load City, FirstValue(Name), as FirstName
from abc.csv group by City;
```

LastValue(*expression*)

Returns the last value in load order of *expression* over a number of records as defined by a **group by** clause. If no text value is found, NULL is returned.

Example:

```
Load City, LastValue(Name), as FirstName
from abc.csv group by City;
```

concat ([**distinct**] *expression*, [*delimiter* [, *sort-weight*]])

returns the aggregated string concatenation of all values of **expression** iterated over a number of records as defined by a *group by* clause. Each value may be separated by the string found in *delimiter*. The order of concatenation may be determined by *sort-weight*. *Sort-weight* should return a numeric value where the lowest value will render the item to be sorted first. If the word **distinct** occurs before the *expression*, all duplicates will be disregarded.

Example:

```
Load Department, concat(Name,',' ) as NameList
from abc.csv group by Department;
```

Counter Aggregation Functions

count([**distinct**] *expression* | *)

Returns the count of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the *expression*, all duplicates will be disregarded.

Occurrences of *expression* returning NULL are disregarded; if **count(*)** is used, however, NULL values are counted as well as non-NULL values.

Examples:

```
Load Month, count(Sales) as
NumberOfSalesPerMonth
from abc.csv
group by Month;

Load Month, count(distinct Customer) as
CustomersBuyingPerMonth
from abc.csv group by Month;

Load Month, count(*) as
NumberOfRecordsPerMonth
from abc.csv group by Month;
```

NumericCount([distinct] expression)

Returns the numeric count of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, NumericCount(Item) as
NumberOfNumericItems from abc.csv
group by Month;
```

TextCount([distinct] expression)

Returns the text count of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, TextCount(Item) as
NumberOfTextItems from abc.csv
group by Month;
```

NullCount([distinct] expression)

Returns the null count of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, NullCount(Item) as
NumberOfNullItems from abc.csv
group by Month;
```

MissingCount([**distinct**] *expression*)

Returns the missing count of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded. Missing values are all non-numeric values, including null values.

Example:

```
Load Month, MissingCount(Item) as  
NumberOfMissingItems from abc.csv  
group by Month;
```

Statistical Aggregation Functions

avg([**distinct**] *expression*)

Returns the average of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, avg(Sales) as  
AverageSalesPerMonth from abc.csv  
group by Month;
```

stdev([**distinct**] *expression*)

Returns the standard deviation of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, stdev(Sales) as  
SalesStandardDeviation from abc.csv group by  
Month;
```

fractile (*expression*, *fractile*)

returns the aggregated fractile of *expression* over a number of records as defined by a **group by** clause.

Example:

```
Load Class, fractile(Grade, 0.75) as F from  
abc.csv group by Class;
```

median (*expression*)

returns the aggregated median of *expression* over a number of records as defined by a **group by** clause.

Example:

```
Load Class, Median(Grade) as MG from abc.csv
group by Class;
```

skew([distinct] expression)

Returns the skewness of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, skew(Sales) as SalesSkew from
abc.csv group by Month;
```

kurtosis([distinct] expression)

Returns the kurtosis of *expression* over a number of records as defined by a **group by** clause. If the word **distinct** occurs before the expression, all duplicates will be disregarded.

Example:

```
Load Month, kurtosis(Sales) as
SalesKurtosis from abc.csv group by Month;
```

correl ([x-expression, y-expression]

returns the aggregated correlation coefficient for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

Example:

```
Load Month, correl(X,Y) as CC from abc.csv
group by Month;
```

sterr ([distinct] expression)

returns the aggregated standard error (stdev/sqrt(n)) for a series of values represented by *expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values will be disregarded. If the word **distinct** occurs before the function arguments, duplicates resulting from the evaluation of the function arguments will be disregarded.

Example:

```
Load Key, sterr(X) as Z from abc.csv
group by Key;
```

steyx (*x-expression*, *y-expression*)

returns the aggregated standard error of the predicted y-value for each x-value in the regression for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

Example:

```
Load Key, steyx(X,Y) as Z from abc.csv
group by Key;
```

linest_m (*x-expression*, *y-expression* [, *y0* [, *x0*]])

returns the aggregated m value (slope) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

Example:

```
Load Key, linest_m(X,Y) as Z from abc.csv
group by Key;
```

linest_b (*x-expression*, *y-expression* [, *y0* [, *x0*]])

returns the aggregated b value (y-intercept) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both $y0$ and $x0$ are stated, the function requires at least two valid data-pairs to calculate. If $y0$ and $x0$ are stated, a single data pair will do.

Example:

```
Load Key, linest_b(X,Y) as Z from abc.csv
group by Key;
```

linest_r2 (*x-expression*, *y-expression* [, $y0$ [, $x0$]])

returns the aggregated r^2 value (coefficient of determination) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value $y0$ may be stated forcing the regression line to pass through the y-axis at a given point. By stating both $y0$ and $x0$ it is possible to force the regression line to pass through a single fixed coordinate.

Unless both $y0$ and $x0$ are stated, the function requires at least two valid data-pairs to calculate. If $y0$ and $x0$ are stated, a single data pair will do.

Example:

```
Load Key, linest_r2(X,Y) as Z from abc.csv
group by Key;
```

linest_sem (*x-expression*, *y-expression* [, $y0$ [, $x0$]])

returns the aggregated standard error of the m value of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value $y0$ may be stated forcing the regression line to pass through the y-axis at a given point. By stating both $y0$ and $x0$ it is possible to force the regression line to pass through a single fixed coordinate.

Unless both y_0 and x_0 are stated, the function requires at least two valid data-pairs to calculate. If y_0 and x_0 are stated, a single data pair will do.

Example:

```
Load Key, linest_sem(X,Y) as Z from abc.csv
group by Key;
```

linest_seb (*x-expression*, *y-expression* [, y_0 [, x_0]])

returns the aggregated standard error of the b value of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value y_0 may be stated forcing the regression line to pass through the y-axis at a given point. By stating both y_0 and x_0 it is possible to force the regression line to pass through a single fixed coordinate.

Unless both y_0 and x_0 are stated, the function requires at least two valid data-pairs to calculate. If y_0 and x_0 are stated, a single data pair will do.

Example:

```
Load Key, linest_seb(X,Y) as Z from abc.csv
group by Key;
```

linest_sey (*x-expression*, *y-expression* [, y_0 [, x_0]])

returns the aggregated standard error of the y estimate of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value y_0 may be stated forcing the regression line to pass through the y-axis at a given point. By stating both y_0 and x_0 it is possible to force the regression line to pass through a single fixed coordinate.

Unless both $y0$ and $x0$ are stated, the function requires at least two valid data-pairs to calculate. If $y0$ and $x0$ are stated, a single data pair will do.

Example:

```
Load Key, linest_sey(X,Y) as Z from abc.csv
group by Key;
```

linest_df (*x-expression*, *y-expression* [, $y0$ [, $x0$]])

returns the aggregated degrees of freedom of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value $y0$ may be stated forcing the regression line to pass through the y-axis at a given point. By stating both $y0$ and $x0$ it is possible to force the regression line to pass through a single fixed coordinate.

Unless both $y0$ and $x0$ are stated, the function requires at least two valid data-pairs to calculate. If $y0$ and $x0$ are stated, a single data pair will do.

Example:

```
Load Key, linest_df(X,Y) as Z from abc.csv
group by Key;
```

linest_f (*x-expression*, *y-expression* [, $y0$ [, $x0$]])

returns the aggregated F statistic ($r^2/(1-r^2)$) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value $y0$ may be stated forcing the regression line to pass through the y-axis at a given point. By stating both $y0$ and $x0$ it is possible to force the regression line to pass through a single fixed coordinate.

Unless both $y0$ and $x0$ are stated, the function requires at least two valid data-pairs to calculate. If $y0$ and $x0$ are stated, a single data pair will do.

Example:

```
Load Key, linest_f(X,Y) as Z from abc.csv
group by Key;
```

linest_ssreg (*x-expression*, *y-expression* [, *y0* [, *x0*]])

returns the aggregated regression sum of squares of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

Example:

```
Load Key, linest_ssreg(X,Y) as Z from abc.csv
group by Key;
```

linest_ssresid (*x-expression*, *y-expression* [, *y0* [, *x0*]])

returns the aggregated residual sum of squares of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over a number of records as defined by a **group by** clause. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

Example:

```
Load Key, linest_ssresid(X,Y) as Z from abc.csv
group by Key;
```

Financial Aggregation Functions

`irr(expression)`

returns the aggregated internal rate of return for a series of cash flows represented by the numbers in *expression* iterated over a number of records as defined by a **group by** clause. These cash flows do not have to be even, as they would be for an annuity. However, the cash flows must occur at regular intervals, such as monthly or annually. The internal rate of return is the interest rate received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods. The function needs at least one positive and one negative value to calculate. Text values, null values and missing values are disregarded.

Example:

```
Load Year, irr(Payments) as IRate from abc.csv
group by Year;
```

`xirr (valueexpression, dateexpression)`

returns the aggregated internal rate of return for a schedule of cash flows (that is not necessarily periodic) represented by paired numbers in *valueexpression* and *dateexpression* iterated over a number of records as defined by a **group by** clause. All payments are discounted based on a 365-day year. Text values, null values and missing values in any or both pieces of a data-pair will result in disregarding the entire data-pair.

Example:

```
Load Year, xirr(Payments, PayDates) as Irate
from abc.csv group by Year;
```

`npv (rate, expression)`

returns the aggregated net present value of an investment based on a discount rate and a series of future payments (negative values) and incomes (positive values) represented by the numbers in *expression* iterated over a number of records as defined by a **group by** clause. The result has a default number format of money. Rate is the interest rate per period. The payments and incomes are assumed to occur at the end of each period. Text values, null values and missing values are disregarded.

Example:

```
Load Year, npv(0.05, Payments) as PValue from
abc.csv group by Year;
```

xnpv (*rate*, *valueexpression*, *dateexpression*)

returns the aggregated net present value for a schedule of cash flows (not necessarily periodic) represented by paired numbers in *valueexpression* and *dateexpression* iterated over a number of records as defined by a **group by** clause. Rate is the interest rate per period. The result has a default number format of money. All payments are discounted based on a 365-day year. Text values, null values and missing values in any or both pieces of a data-pair will result in disregarding the entire data-pair.

Example:

```
Load Year, npv(0.05, Payments, PayDates) as
PValue from abc.csv group by Year;
```

Statistical Test Functions

The following three functions apply to χ^2 -tests.

chi2test_p (*col*, *row*, *observed_value* [, *expected_value*])

returns the aggregated χ^2 -test p value (significance) for one or two series of values iterated over a number of records as defined by a **group by** clause. The test can be done either on the values in *observed_value* testing for variations within the specified *col* and *row* matrix or by comparing values in *observed_value* with corresponding values in *expected_values*. Text values, null values and missing values in the value expressions will result in the function returning null

Example:

```
Load Year, chi2test_p
(Gender, Description, Observed, Expected)
as X from abc.csv group by Year;
```

chi2test_df (*col*, *row*, *observed_value* [, *expected_value*])

returns the aggregated χ^2 -test df value (degrees of freedom) for one or two series of values iterated over a number of records as defined by a **group by** clause. See the **chi2test_p** function, page 372, for argument descriptions.

Example:

```
Load Year, chi2test_df
(Gender, Description, Observed, Expected)
as X from abc.csv group by Year;
```

chi2test_chi2 (*col, row, observed_value* [, *expected_value*])

returns the aggregated chi²-test value for one or two series of values iterated over a number of records as defined by a **group by** clause. See the **chi2test_p** function, page 372, for argument descriptions.

Example:

```
Load Year,
chi2test_chi2(Gender,Description,Observed,
Expected) as X from abc.csv group by Year;
```

The following eight functions apply to two independent samples' student's t-tests.

TTest_t (*group, value* [, *eq_var = true*])

returns the aggregated student's t-test t value for two independent series of values iterated over a number of records as defined by a **group by** clause. The values should be returned by *value* and be logically grouped via exactly two values in *group*. If *eq_var* is specified as false, separate variances of the two samples will be assumed, else equal variances between the samples will be assumed. Text values, null values and missing values in *value* will result in the function returning null.

Example:

```
Load Year, ttest_t(Group, Value) as X
from abc.csv
group by Year;
```

TTest_df (*group, value* [, *eq_var = true*])

returns the aggregated student's t-test df value (degrees of freedom) for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTest_t** function, page 373, for argument descriptions.

Example:

```
Load Year, ttest_df(Group, Value) as X from
abc.csv group by Year;
```

TTest_sig (*group, value* [, *eq_var = true*])

returns the aggregated student's t-test 2-tailed level of significance for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTest_t** function, page 373, for argument descriptions.

Example:

```
Load Year, ttest_sig(Group, Value) as X from  
abc.csv group by Year;
```

TTest_dif (*group, value [, eq_var = true]*)

returns the aggregated student's t-test mean difference for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTest_t** function, page 373, for argument descriptions.

Example:

```
Load Year, ttest_dif(Group, Value) as X from  
abc.csv group by Year;
```

TTest_sterr (*group, value [, eq_var = true]*)

returns the aggregated student's t-test standard error of the mean difference for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTest_t** function, page 373, for argument descriptions.

Example:

```
Load Year, ttest_sterr(Group, Value) as X from  
abc.csv group by Year;
```

TTest_conf (*group, value [, sig = 0.025 [, eq_var = true]]*)

returns the aggregated t value for two independent series of values iterated over a number of records as defined by a **group by** clause. The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTest_t** function, page 373, for descriptions of the other arguments.

Example:

```
Load Year, ttest_conf(Group, Value) as X from  
abc.csv group by Year;
```

TTest_lower (*group, value [, sig = 0.025 [, eq_var = true]]*)

returns the aggregated value for the lower end of the confidence interval for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTest_t**, page 373, and **TTest_conf**, page 374, functions for argument descriptions.

Example:

```
Load Year, ttest_lower(Group, Value) as X from
abc.csv group by Year;
```

TTest_upper (group, value [, sig = 0.025 [, eq_var = true]])

returns the aggregated value for the upper end of the confidence interval for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTest_t**, page 373, and **TTest_conf**, page 374, functions for argument descriptions.

Example:

```
Load Year, ttest_upper(Group, Value) as X from
abc.csv group by Year;
```

The following eight functions apply to two independent samples' student's t-tests where the input data series is given in weighted two-column format.

TTestw_t (weight, group, value [, eq_var = true])

returns the aggregated student's t-test t value for two independent series of values iterated over a number of records as defined by a **group by** clause. The values should be returned by *value* and be logically grouped via exactly two values in *group*. Each value in *value* can be counted one or more times according to a corresponding weight value in *weight*. If *eq_var* is specified as false, separate variances of the two samples will be assumed, else equal variances between the samples will be assumed. Text values, null values and missing values in *value* will result in the function returning null.

Example:

```
Load Year, ttestw_t(Weight, Group, Value) as X
from abc.csv group by Year;
```

TTestw_df (weight, group, value [, eq_var = true])

returns the aggregated student's t-test df value (degrees of freedom) for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTestw_t** function, page 375, for argument descriptions.

Example:

```
Load Year, ttestw_df(Weight, Group, Value) as X
from abc.csv group by Year;
```

TTestw_sig (*weight, group, value* [, *eq_var = true*])

returns the aggregated student's t-test 2-tailed level of significance for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTestw_t** function, page 375, for argument descriptions.

Example:

```
Load Year, ttestw_sig(Weight, Group, Value) as
X from abc.csv group by Year;
```

TTestw_dif (*weight, group, value* [, *eq_var = true*])

returns the aggregated student's t-test mean difference for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTestw_t** function, page 375, for argument descriptions.

Example:

```
Load Year, ttestw_dif(Weight, Group, Value) as
X from abc.csv group by Year;
```

TTestw_sterr (*weight, group, value* [, *eq_var = true*])

returns the aggregated student's t-test standard error of the mean difference for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTestw_t** function, page 375, for argument descriptions.

Example:

```
Load Year, ttestw_sterr(Weight, Group, Value)
as X from abc.csv group by Year;
```

TTestw_conf (*weight, group, value* [, *sig = 0.025* [, *eq_var = true*]])

returns the aggregated t value for two independent series of values iterated over a number of records as defined by a **group by** clause. The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTestw_t** function, page 375, for descriptions of the other arguments.

Example:

```
Load Year, ttestw_conf(Weight, Group, Value)
as X from abc.csv group by Year;
```

TTestw_lower (*weight, group, value* [, *sig = 0.025* [, *eq_var = true*]])

returns the aggregated value for the lower end of the confidence interval for two independent series of values iterated over a number

of records as defined by a **group by** clause. See the **TTestw_t**, page 375, and **TTestw_conf**, page 376, functions for argument descriptions.

Example:

```
Load Year, ttestw_lower(Weight, Group, Value)
as X from abc.csv group by Year;
```

TTestw_upper (*weight, group, value* [, *sig = 0.025* [, *eq_var = true*]])

returns the aggregated value for the upper end of the confidence interval for two independent series of values iterated over a number of records as defined by a **group by** clause. See the **TTestw_t**, page 375, and **TTestw_conf**, page 376, functions for argument descriptions.

Example:

```
Load Year, ttestw_upper(Weight, Group, Value)
as X from abc.csv group by Year;
```

The following eight functions apply to one-sample student's t-tests.

TTest1_t (*value*)

returns the aggregated t value for a series of values iterated over a number of records as defined by a **group by** clause. The values should be returned by *value*. Text values, null values and missing values in *value* will result in the function returning null.

Example:

```
Load Year, ttest1_t(Value) as X from abc.csv
group by Year;
```

TTest1_df (*value*)

returns the aggregated student's t-test df value (degrees of freedom) for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1_t** function, page 377, for argument descriptions.

Example:

```
Load Year, ttest1_df(Value) as X from abc.csv
group by Year;
```

TTest1_sig (*value*)

returns the aggregated student's t-test 2-tailed level of significance for a series of values iterated over a number of records as defined by

a **group by** clause. See the **TTest1_t** function, page 377, for argument descriptions.

Example:

```
Load Year, ttest1_sig(Value) as X from abc.csv
group by Year;
```

TTest1_dif (*value*)

returns the aggregated student's t-test mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1_t** function, page 377, for argument descriptions.

Example:

```
Load Year, ttest1_dif(Value) as X from abc.csv
group by Year;
```

TTest1_sterr (*value*)

returns the aggregated student's t-test standard error of the mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1_t** function, page 377, for argument descriptions.

Example:

```
Load Year, ttest1_sterr(Value) as X from
abc.csv group by Year;
```

TTest1_conf (*value* [, *sig* = 0.025])

returns the aggregated t value for a series of values iterated over a number of records as defined by a **group by** clause. The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTest1_t** function, page 377, for descriptions of the other arguments.

Example:

```
Load Year, ttest1_conf(Value) as X from abc.csv
group by Year;
```

TTest1_lower (*value* [, *sig* = 0.025])

returns the aggregated value for the lower end of the confidence interval for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1_t**, page 377, and **TTest1_conf**, page 378, functions for argument descriptions.

Example:

```
Load Year, ttest1_lower(Value) as X from
abc.csv group by Year;
```

TTest1_upper (value [, sig = 0.025])

returns the aggregated value for the upper end of the confidence interval for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1_t**, page 377, and **TTest1_conf**, page 378, functions for argument descriptions.

Example:

```
Load Year, ttest1_upper(Value) as X from
abc.csv group by Year;
```

The following eight functions apply to one-sample student's t-tests where the input data series is given in weighted two-column format.

TTest1w_t (weight, value)

returns the aggregated t value for a series of values iterated over a number of records as defined by a **group by** clause. The values should be returned by *value*. Each value in *value* can be counted one or more times according to a corresponding weight value in *weight*. Text values, null values and missing values in *value* will result in the function returning null.

Example:

```
Load Year, ttest1w_t(Weight, Value) as X from
abc.csv group by Year;
```

TTest1w_df (weight, value)

returns the aggregated student's t-test df value (degrees of freedom) for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1w_t** function, page 379, for argument descriptions.

Example:

```
Load Year, ttest1w_df(Weight, Value) as X from
abc.csv group by Year;
```

TTest1w_sig (weight, value)

returns the aggregated student's t-test 2-tailed level of significance for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1w_t** function, page 379, for argument descriptions.

Example:

```
Load Year, ttest1w_sig(Weight, Value) as X from
abc.csv group by Year;
```

TTest1w_dif (*weight, value*)

returns the aggregated student's t-test mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1w_t** function, page 379, for argument descriptions.

Example:

```
Load Year, ttest1w_dif(Weight, Value) as X from
abc.csv group by Year;
```

TTest1w_sterr (*weight, value*)

returns the aggregated student's t-test standard error of the mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1w_t** function, page 379, for argument descriptions.

Example:

```
Load Year, ttest1w_sterr(Weight, Value) as X
from abc.csv group by Year;
```

TTest1w_conf (*weight, value* [, *sig* = 0.025])

returns the aggregated t value for a series of values iterated over a number of records as defined by a **group by** clause. The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTest1w_t** function, page 379, for descriptions of the other arguments.

Example:

```
Load Year, ttest1w_conf(Weight, Value) as X
from abc.csv group by Year;
```

TTest1w_lower (*weight, value* [, *sig* = 0.025])

returns the aggregated value for the lower end of the confidence interval for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1w_t**, page 379, and **TTest1w_conf**, page 380, functions for argument descriptions.

Example:

```
Load Year, ttest1w_lower(Weight, Value) as X
from abc.csv group by Year;
```

TTest1w_upper (*weight*, *value* [, *sig* = 0.025])

returns the aggregated value for the upper end of the confidence interval for a series of values iterated over a number of records as defined by a **group by** clause. See the **TTest1w_t**, page 379, and **TTest1w_conf**, page 380, functions for argument descriptions.

Example:

```
Load Year, ttest1w_upper(Weight, Value) as X
from abc.csv group by Year;
```

The following five functions apply to z-tests.

ZTest_z (*value* [, *sigma*])

returns the aggregated z-test t value for a series of values iterated over a number of records as defined by a **group by** clause. The values should be returned by *value*. A population mean of 0 is assumed. If you want the test to be performed around another mean, subtract that value from the sample values. If known, the standard deviation can be stated in *sigma*. If *sigma* is omitted the actual sample standard deviation will be used. Text values, null values and missing values in *value* will result in the function returning null.

Example:

```
Load Year, ztest_z(Value-TestValue) as X from
abc.csv group by Year;
```

ZTest_sig (*value* [, *sigma*])

returns the aggregated z-test 2-tailed level of significance for a series of values iterated over a number of records as defined by a **group by** clause. See the **ZTest_z** function, page 381, for argument descriptions.

Example:

```
Load Year, ztest_sig(Value-TestValue) as X from
abc.csv group by Year;
```

ZTest_dif (*value* [, *sigma*])

returns the aggregated z-test mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **ZTest_t** function, page 381, for argument descriptions.

Example:

```
Load Year, ztest_dif(Value-TestValue) as X from
abc.csv group by Year;
```

ZTest_sterr (*value* [, *sigma*])

returns the aggregated z-test standard error of the mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **ZTest_z** function, page 381, for argument descriptions.

Example:

```
Load Year, ztest_sterr(Value-TestValue) as X
from abc.csv group by Year;
```

ZTest_conf (*value* [, *sigma* [, *sig* = 0.025]])

returns the aggregated z-test t value for a series of values iterated over a number of records as defined by a **group by** clause. The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **ZTest_z** function, page 381, for descriptions of the other arguments.

Example:

```
Load Year, ztest_conf(Value-TestValue) as X
from abc.csv group by Year;
```

The following five functions apply to z-tests where the input data series is given in weighted two-column format.

ZTestw_z (*weight*, *value* [, *sigma*])

returns the aggregated z value for a series of values iterated over a number of records as defined by a **group by** clause. The values should be returned by *value*. A sample mean of 0 is assumed. If you want the test to be performed around another mean, subtract that value from the sample values. Each value in *value* can be counted one or more times according to a corresponding weight value in *weight*. If known, the standard deviation can be stated in *sigma*. If *sigma* is omitted the actual sample standard deviation will be used. Text values, null values and missing values in *value* will result in the function returning null.

Example:

```
Load Year, ztestw_t(Weight,Value-TestValue) as
X from abc.csv group by Year;
```

ZTestw_sig (*weight*, *value* [, *sigma*])

returns the aggregated z-test 2-tailed level of significance for a series of values iterated over a number of records as defined by a **group by** clause. See the **ZTestw_z** function, page 382, for argument descriptions.

Example:

```
Load Year, ztestw_sig(Weight,Value-TestValue)
as X from abc.csv group by Year;
```

ZTestw_dif (*weight, value* [, *sigma*])

returns the aggregated z-test mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **ZTestw_z** function, page 382, for argument descriptions.

Example:

```
Load Year, ztestw_dif(Weight,Value-TestValue)
as X from abc.csv group by Year;
```

ZTestw_sterr (*weight, value* [, *sigma*])

returns the aggregated z-test standard error of the mean difference for a series of values iterated over a number of records as defined by a **group by** clause. See the **ZTestw_z** function, page 382, for argument descriptions.

Example:

```
Load Year, ztestw_sterr(Weight,Value-TestValue)
as X from abc.csv group by Year;
```

ZTestw_conf (*weight, value* [, *sigma*] [, *sig* = 0.025])

returns the aggregated z value for a series of values iterated over a number of records as defined by a **group by** clause. The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **ZTestw_z** function, page 382, for descriptions of the other arguments.

Example:

```
Load Year, ztestw_conf(Weight,Value-TestValue)
as X from abc.csv group by Year;
```

22.4 Other Functions

In addition to the aggregation functions, QlikView offers more than 100 standard functions grouped as follows:

General numeric functions	page 384
Range functions	page 388
Exponential and logarithmic functions.....	page 394
Trigonometric and hyperbolic functions	page 394

Statistical distribution functions	page 395
Financial functions	page 397
Mathematical constants and parameter-free functions	page 402
Counter functions	page 402
String functions	page 405
Mapping functions.....	page 411
Inter-record functions	page 412
Conditional functions	page 415
Logical functions.....	page 417
Null functions	page 418
System functions	page 418
File functions.....	page 421
Table functions	page 424
Document functions	page 426
Date and time functions.....	page 426
Number interpretation functions	page 449
Formatting functions	page 453
Color functions.....	page 458

General Numeric Functions

div($x1$, $x2$)

Integer division. Both parameters are interpreted as real numbers, i.e. they do not have to be integers. The result is the integer part of the result of the arithmetic division, a real number.

Examples:

div(7,2) returns 3

div(9,3) returns 3

div(-4,3) returns -1

div(4,-3) returns -1

div(-4,-3) returns 1

mod($x1$, $x2$)

Mathematical modula function. Both parameters must have integer values. $x2$ must be greater than 0. The result is the mathematical

modula function, i.e. the non-negative remainder of an integer division.

Examples:

mod(7,2) returns 1
mod(9,3) returns 0
mod(-4,3) returns 2
mod(4,-3) returns NULL
mod(-4,-3) returns NULL

fmod(*x1* , *x2*)

Generalized modula function. Both parameters are interpreted as real numbers, i.e. they do not have to be integers. The result is the remainder of the integer division, a real number.

Examples:

fmod(7,2) returns 1
fmod(9,3) returns 0
fmod(-4,3) returns -1
fmod(4,-3) returns 1
fmod(-4,-3) returns -1

ceil(*x* [, *base* [, *offset*]])

Rounding of *x* upwards to the nearest multiple of *base* with an offset of *offset*. The result is a number.

Examples:

ceil(2.4) returns 3
ceil(2.6) returns 3
ceil(3.88 , 0.1) returns 3.9
ceil(3.88 , 5) returns 5
ceil(1.1 , 1 , 0.5) returns 1.5

floor(*x* [, *base* [, *offset*]])

Rounding of *x* downwards to the nearest multiple of *base* with an offset of *offset*. The result is a number.

Examples:

floor(2.4) returns 2
floor(2.6) returns 2
floor(3.88 , 0.1) returns 3.8
floor(3.88 , 5) returns 0
floor(1.1 , 1 , 0.5) returns 0.5

frac(*x*)

Returns the fraction part of *x*. The fraction is defined in such a way that **frac(*x*)+floor(*x*)=x**.

Examples:

frac(11.43) returns 0.43

frac(-1.4) returns 0.6

round(*x* [, *step* [, *offset*]])

Rounding of *x* upwards or downwards *n* number of steps with an offset of *offset*. The result is a number. If *x* is exactly in the middle of an interval, it is rounded upwards. The function complies with the IEEE standard 64-bit floating point numbers.

Examples:

round(2.4) returns 2

round(2.6) returns 3

round(2.5) returns 3

round(3.88 , 0.1) returns 3.9

round(3.88 , 5) returns 5

round(1.1 , 1 , 0.5) returns 1.5

fabs(*x*)

The absolute value of *x*. The result is a positive number.

Examples:

fabs(2.4) returns 2.4

fabs(-3.8) returns 3.8

sign(*x*)

Returns 1, 0, or -1 depending on whether *x* is a positive number, 0, or a negative number. If no numeric value is found, NULL is returned.

Examples:

sign(66) returns 1

sign(0) returns 0

sign(-234) returns -1

numsum(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangesum**.

numcount(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangenumericcount**.

numavg(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangeavg**.

nummin(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangemin**.

nummax(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangemax**.

fact(*n*)

Returns the factorial of a positive integer *n*. If the number *n* is not an integer it will be truncated. Non-positive numbers will return NULL.

Examples:

fact(1) returns 1

fact(5) returns 120 (1*2*3*4*5 = 120)

fact(-5) returns NULL

combin(*n1*, *n2*)

Returns the number of combinations of *n2* items that can be picked from a group of *n1* items. The order in which the items are picked is not significant. Non-integer arguments will be truncated.

Example:

How many combinations of seven numbers can be picked from a total of 35 lotto numbers?

combin(35,7) returns 6 724 520

permut(*n1*, *n2*)

Returns the number of permutations of *n2* items that can be selected from a group of *n1* items. The order in which the items are picked is significant. Non-integer arguments will be truncated.

Example:

In how many ways could the gold, silver and bronze medals be distributed after a 100m final with 8 participants?

permut(8,3) returns 336

even(*n*)

Returns true if *n* is an even integer, false if *n* is an odd integer and NULL if *n* is not an integer.

Examples:

even(3) returns false

even(2*10) returns true
even(3.14) returns NULL

odd(*n*)

Returns true if *n* is an odd integer, false if *n* is an even integer and NULL if *n* is not an integer.

Examples:

odd(3) returns true
odd(2*10) returns false
odd(3.14) returns NULL

BitCount(*i*)

Returns the number of set bits in *i*. *I* will be interpreted as a signed 64-bit integer, though only the first 48 bits are significant. A value above 48-bits will return NULL.

Examples:

bitcount(3) returns 2
bitcount(100) returns 3
bitcount(*n*)+bitcount(bitnot *n*) returns 64 (as long as *n* is a valid integer)

Range Functions

rangesum(*expr1* [, *expr2*, ... *exprN*])

Returns the sum of a range of 1 to *N* arguments. As opposed to the + operator, **rangesum** will treat all non-numeric values as 0.

Examples:

rangesum(1,2,4) returns 7
rangesum(1,'xyz') returns 1
rangesum(null()) returns 0

rangeavg(*expr1* [, *expr2*, ... *exprN*])

Returns the average of a range of 1 to *N* arguments. If no numeric value is found, NULL is returned.

Examples:

rangeavg(1,2,4) returns 2.33333333
rangeavg(1,'xyz') returns 1
rangeavg(null(), 'abc') returns NULL

rangecount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of values found within a range of 1 to N arguments. Null values are not counted.

Examples:

rangecount(1,2,4) returns 3

rangecount(2,'xyz') returns 2

rangecount(null()) returns 0

rangecorrel(*x-value*, *y-value* {,*x-value*, *y-value*})

returns correlation coefficient for a series of coordinates.

X-value is a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 412).

Y-value is a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 412). Each value or range of values must correspond to an *x-value* or a range of *x-values*.

The function needs at least two pairs of coordinates to calculate.

Text values, null values and missing values are disregarded.

Examples:

rangecorrel (2,3,6,8,9,4) returns 0,269

rangecorrel (above(sum(X), 0, 10), above(Y, 0, 10))

rangecorrel (above(total X,0,rowno(total)),

above(total Y,0,rowno(total)))

rangemin(*expr1* [, *expr2*, ... *exprN*])

Returns the lowest numeric value found within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

Examples:

rangemin(1,2,4) returns 1

rangemin(1,'xyz') returns 1

rangemin(null(), 'abc') returns NULL

rangemax(*expr1* [, *expr2*, ... *exprN*])

Returns the highest numeric values found within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

Examples:

rangemax(1,2,4) returns 4

rangemax(1,'xyz') returns 1

rangemax(null(), 'abc') returns NULL

rangestdev(*expr1* [, *expr2*, ... *exprN*])

Returns the standard deviation within a range of 1 to N arguments.
If no numeric value is found, NULL is returned.

Examples:

rangestdev(1,2,4) returns 1.5275252316519

rangestdev(null()) returns NULL

rangeskew(*expr1* [, *expr2*, ... *exprN*])

Returns the skewness within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

Example:

rangeskew(1,2,4) returns 0.93521952958283

rangekurtosis(*expr1* [, *expr2*, ... *exprN*])

Returns the kurtosis within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

Example:

rangekurtosis(1,2,4,7) returns -0.28571428571429

rangefractile(*fractile*, *expr1* [, *expr2*, ... *exprN*])

Returns the fractile within a range of 1 to N arguments.

Example:

rangefractile(0.25,1,2,4,6) returns 1.5

rangenumericcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of numeric values found within a range of 1 to N arguments.

Examples:

rangenumericcount(1,2,4) returns 3

rangenumericcount(2,'xyz') returns 1

rangenumericcount(null()) returns 0

rangetextcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of text values found within a range of 1 to N arguments.

Examples:

rangetextcount(1,2,4) returns 0

rangetextcount(2,'xyz') returns 1

rangetextcount(null()) returns 0

rangenullcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of numeric values found within a range of 1 to N arguments.

Examples:

rangenullcount (1,2,4) returns 0

rangenullcount (2,'xyz') returns 0

rangenullcount (null(),null()) returns 2

rangemissingcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of non-numeric values (including null values) found among 1 to N arguments.

Examples:

rangemissingcount (1,2,4) returns 0

rangemissingcount (2,'xyz') returns 1

rangemissingcount (null()) returns 1

rangemode(*expr1* [, *expr2*, ... *exprN*])

Returns the mode value, i.e. the most commonly occurring value within a range of 1 to N arguments. If more than one value shares the highest frequency, NULL is returned.

Examples:

rangemode (1,2,9,2,4) returns 2

rangemode ('a',4,'a',4) returns NULL

rangemode (null()) returns NULL

rangeonly(*expr1* [, *expr2*, ... *exprN*])

If exactly one non-null value exists among the range of N expressions, that value will be returned. In all other cases, NULL is returned.

Examples:

rangeonly (1,2,4) returns NULL

rangeonly (1,'xyz') returns NULL

rangeonly (null(), 'abc') returns 'abc'

rangeminstring(*expr1* [, *expr2*, ... *exprN*])

Returns the first text value in text sort order found among 1 to N arguments.

Examples:

rangeminstring (1,2,4) returns 1

rangeminstring ('xyz','abc') returns 'abc'

rangeminstring (null()) returns NULL

rangemaxstring(*expr1* [, *expr2*, ... *exprN*])

Returns the last text value in text sort order found among 1 to N arguments.

Examples:

rangemaxstring (1,2,4) returns 4

rangemaxstring ('xyz','abc') returns 'xyz'

rangemaxstring (null()) returns NULL

rangeirr(*value* {,*value*})

returns the internal rate of return for a series of cash flows represented by the numbers in values. These cash flows do not have to be even, as they would be for an annuity. However, the cash flows must occur at regular intervals, such as monthly or annually. The internal rate of return is the interest rate received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods.

Value is a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 412). The function needs at least one positive and one negative value to calculate. Text values, null values and missing values are disregarded.

Examples:

rangeirr(-70000,12000,15000,18000,21000,26000) returns 0,0866

rangeirr(**above**(**sum**(value), 0, 10))

rangeirr(**above**(**total** value, 0, **rowno**(**total**)))

rangenpv (*rate*, *value* {,*value*})

returns the net present value of an investment based on a discount rate and a series of future payments (negative values) and incomes (positive values). The result has a default number format of money.

Rate is the interest rate per period.

Value is a payment or income occurring at the end of each period. Each *value* may be a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 412). Text values, null values and missing values are disregarded.

Examples:

rangenpv(0.1,-10000,3000,4200,6800) returns 1188,44

rangenpv(0.05, **above**(**sum**(value), 0, 10))

rangenpv(0.05, **above**(**total** value, 0, **rowno**(**total**)))

rangexirr(*value, date {,value, date}*)

returns the internal rate of return for a schedule of cash flows that is not necessarily periodic. To calculate the internal rate of return for a series of periodic cash flows, use the **rangeirr** function.

Value is a cash flow or a series of cash flows that corresponds to a schedule of payments in dates. Each *value* may be a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 412). Text values, null values and missing values are disregarded. All payments are discounted based on a 365-day year. The series of values must contain at least one positive and one negative value.

Date is a payment date or a schedule of payment dates that corresponds to the cash flow payments.

Examples:

rangexirr(-2500,'2008-01-01',2750,'2008-09-01') returns 0,1532

rangexirr (**above**(**sum**(value), 0, 10), **above**(date, 0, 10))

rangexirr(**above**(**total** value,0,**rowno**(**total**)),

above(**total** date,0,**rowno**(**total**)))

rangexnpv(*rate, value, date {,value, date}*)

Returns the net present value for a schedule of cash flows that is not necessarily periodic. The result has a default number format of money. To calculate the net present value for a series of periodic cash flows, use the **rangenv** function.

Rate is the interest rate per period.

Value is a cash flow or a series of cash flows that corresponds to a schedule of payments in dates. Each *value* may be a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 412). Text values, null values and missing values are disregarded. All payments are discounted based on a 365-day year. The series of values must contain at least one positive and one negative value.

Date is a payment date or a schedule of payment dates that corresponds to the cash flow payments.

Examples:

rangexnpv(0.1, -2500,'2008-01-01',2750,'2008-09-01') returns 80,25

rangexnpv (0.1, **above**(**sum**(value), 0, 10), **above**(date, 0, 10))

rangexnpv(0.1, **above**(**total** value,0,**rowno**(**total**)),

above(total date,0,rowno(total)))

Exponential and Logarithmic Functions

exp(x)

Exponential function, with the base of the natural logarithm e as base. The result is a positive number.

log(x)

The natural logarithm of x . The function is only defined if $x > 0$. The result is a number.

log10(x)

The 10-logarithm (base 10) of x . The function is only defined if $x > 0$. The result is a number.

sqrt(x)

Square root of x . The function is only defined if $x \geq 0$. The result is a positive number.

sqr(x) Square of x . The result is a number.

pow(x,y)

Returns x to the power of y . The result is a number.

Trigonometric and Hyperbolic Functions

All angles are measured in radians.

cos(x)

Cosine of x . The result is a number between -1 and 1.

acos(x)

Arcus cosine of x . The function is only defined if $-1 \leq x \leq 1$. The result is a number between 0 and π .

sin(x)

Sine of x . The result is a number between -1 and 1.

asin(x)

Arcus sine of x . The function is only defined if $-1 \leq x \leq 1$. The result is a number between $-\pi/2$ and $\pi/2$.

tan(x)

Tangent of x . The result is a number.

atan(x)

Arcus tangent of x . The result is a number between $-\pi/2$ and $\pi/2$.

atan2(*y,x*)

Two-dimensional generalization of the arcus tangent function.

Returns the angle between the origin and the point represented by the coordinates *x* and *y*. The result is a number between $-\pi$ and $+\pi$.

cosh(*x*)

Cosine hyperbolicus of *x*. The result is a positive number.

sinh(*x*)

Sine hyperbolicus of *x*. The result is a number.

tanh(*x*)

Tangent hyperbolicus of *x*. The result is a number.

Statistical Distribution Functions

The statistical distribution functions described below are all implemented in QlikView using the Cephes library. For references and details on algorithms used, accuracy etc., please see <http://www.netlib.org/cephes/>. The Cephes function library is used by permission.

chidist (*value, degrees_freedom*)

returns the one-tailed probability of the chi-squared distribution.

The χ^2 distribution is associated with a χ^2 test. *Value* is the value at which you want to evaluate the distribution. *Value* must not be negative. *Degrees_freedom* is a positive integer stating the number of degrees of freedom. Both arguments must be numeric, else null will be returned. This function is related to the **chiinv** function in the following way:

If *prob* = **chidist**(*value,df*), then **chiinv**(*prob, df*) = *value*.

Example:

chidist(8, 15) returns 0.9237827

chiinv (*prob, degrees_freedom*)

returns the inverse of the one-tailed probability of the chi-squared distribution. *Prob* is a probability associated with the chi-squared distribution. It must be a number between 0 and 1. *Degrees_freedom* is an integer stating the number of degrees of freedom. Both arguments must be numeric, else null will be returned. This function is related to the **chidist** function in the following way:

If *prob* = **chidist**(*value,df*), then **chiinv**(*prob, df*) = *value*.

Example:

chiinv(0.9237827, 15) returns 8.0000001

normdist (*value, mean, standard_dev*)

returns the cumulative normal distribution for the specified mean and standard deviation. *Value* is the value at which you want to evaluate the distribution. *Mean* is a value stating the arithmetic mean for the distribution. *Standard_dev* is a positive value stating the standard deviation of the distribution. All arguments must be numeric, else null will be returned. If *mean* = 0 and *standard_dev* = 1, the function returns the standard normal distribution. This function is related to the **norminv** function in the following way:

If $prob = \text{normdist}(value, m, sd)$, then $\text{norminv}(prob, m, sd) = value$.

Example:

chdist(0.5, 0, 1) returns 0.6914625

norminv (*prob, mean, standard_dev*)

returns the inverse of the normal cumulative distribution for the specified mean and standard deviation. *Prob* is a probability associated with the normal distribution. It must be a number between 0 and 1. *Mean* is a value stating the arithmetic mean for the distribution. *Standard_dev* is a positive value stating the standard deviation of the distribution. All arguments must be numeric, else null will be returned. This function is related to the **normdist** function in the following way:

If $prob = \text{normdist}(value, m, sd)$, then $\text{norminv}(prob, m, sd) = value$.

Example:

norminv(0.6914625, 0, 1) returns 0.4999717

tdist (*value, degrees_freedom, tails*)

returns the probability for the Student t-distribution where a numeric value is a calculated value of t for which the probability is to be computed. *Value* is the value at which you want to evaluate the distribution and must not be negative. *Degrees_freedom* is a positive integer stating the number of degrees of freedom. *Tails* must be either 1 (one-tailed distribution) or 2 (two-tailed distribution). All arguments must be numeric, else null will be returned. This function is related to the **tinvs** function in the following way:

If $prob = \text{tdist}(value, df, 2)$, then $\text{tinvs}(prob, df) = value$.

Example:

chdist(1, 30, 2) returns 0.3253086

tin_v (*prob*, *degrees_freedom*)

returns the t-value of the Student's t-distribution as a function of the probability and the degrees of freedom. *Prob* is a two-tailed probability associated with the t-distribution. It must be a number between 0 and 1. *Degrees_freedom* is an integer stating the number of degrees of freedom. Both arguments must be numeric, else null will be returned. This function is related to the **tdist** function in the following way:

If $prob = \mathbf{tdist}(value, df, 2)$, then $\mathbf{tin}_v(prob, df) = value$.

Example:

tin_v(0.3253086, 30) returns 1

fdist (*value*, *degrees_freedom1*, *degrees_freedom2*)

returns the F-distribution.. *Value* is the value at which you want to evaluate the distribution. *Value* must not be negative. *Degrees_freedom1* is a positive integer stating the number of numerator degrees of freedom. *Degrees_freedom2* is a positive integer stating the number of denominator degrees of freedom. All arguments must be numeric, else null will be returned. This function is related to the **finv** function in the following way:

If $prob = \mathbf{fdist}(value, df1, df2)$, then $\mathbf{fin}_v(prob, df1, df2) = value$.

Example:

fdist(15, 8, 6) returns 0.0019369

finv (*prob*, *degrees_freedom1*, *degrees_freedom2*)

returns the inverse of the F-distribution. *Prob* is a probability associated with the F-distribution and must be a number between 0 and 1. *Degrees_freedom* is an integer stating the number of degrees of freedom. All arguments must be numeric, else null will be returned. This function is related to the **fdist** function in the following way: If $prob = \mathbf{fdist}(value, df1, df2)$, then $\mathbf{fin}_v(prob, df1, df2) = value$.

Example:

finv(0.0019369, 8, 5) returns 15.0000197

Financial Functions

Note When using the financial functions described below, it is vital to be consistent about the units you use for specifying *rate* and *nper*. If you make monthly payments on a five-year loan at 6% annual interest, use 0.005 (6%/12) for *rate* and 60 (5*12) for *nper*. If you make annual payments on the same loan, use 6% for *rate* and 5 for *nper*. In all the functions, cash you pay

out is represented by negative numbers. Cash you receive is represented by positive numbers.

BlackAndSchole(*strike, time_left, underlying_price, vol, risk_free_rate, type*)

Returns the theoretical value of an option according to Black and Schole's unmodified formula (European style options).

Strike is the future purchase price of the stock.

Time_left is the number of time periods remaining.

Underlying_price is the current value of the stock.

Vol is the volatility in % per time period.

Risk_free_rate is the risk free rate in % per time period.

Type is 'c', 'call' or any non-zero numeric value for call-options and 'p', 'put' or 0 for put-options..

Example:

The theoretical price of an option to buy in 4 years at a value of 130 a share which is today worth 68.5 assuming a volatility of 40% per year and a risk-free interest rate of 4% ?

blackandschole(130, 4, 68.5, 0.4, 0.04, 'call') returns 11.245...

The following approximation is used to calculate the inverse of the normal cumulative distribution in the Black and School formula:

For $x > 0$

$$\int_{-\infty}^x e^{-u^2/2} du \approx \sqrt{2\pi} - (b_1 k + b_2 k^2 + b_3 k^3) e^{-x^2/2}$$

$$k = \frac{1}{1 + 0.33267x} \quad b = \begin{pmatrix} 0.4361836 \\ -0.1201676 \\ 0.937298 \end{pmatrix}$$

For $x < 0$

$$f(x) = 1 - f(-x).$$

fv(*rate*, *nper*, *pmt* [, *pv* [, *type*]])

Returns the future value of an investment based on periodic, constant payments and a constant interest rate. The result has a default number format of money.

Rate is the interest rate per period.

Nper is the total number of payment periods in an annuity.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *pv* argument.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now. If *pv* is omitted, it is assumed to be 0 (zero), and you must include the *pmt* argument.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Example:

You pay a new VCR by 36 monthly installments of \$20. The interest rate is 6% per annum. The bill comes at the end of the month. What is the total value of the money you invested when the last bill has been paid?

fv(0.005,36,-20) returns \$786.72

nper(*rate*, *pmt*, *pv* [, *fv* [, *type*]])

Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.

Rate is the interest rate per period.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *pv* argument.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, it is assumed to be 0.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Example:

You want to sell a VCR by monthly installments of \$20. The interest rate is 6% per annum. The bill comes at the end of the month. How

many periods are required if the value of the money you have received after the last bill has been paid should equal \$786.72?

nper(0.005,-20,0,786.72) returns 36

pmt(*rate*, *nper*, *pv* [, *fv* [, *type*]])

Returns the payment for a loan based on constant payments and a constant interest rate. The result has a default number format of money.

Rate is the interest rate per period.

Nper is the total number of payment periods in an annuity.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, it is assumed to be 0.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

To find the total amount paid over the duration of the loan, multiply the returned pmt value by nper.

Examples:

The following formula returns the monthly payment on a \$20,000 loan at an annual rate of 10 percent that you must pay off in 8 months:

pmt(0.1/12, 8, 20000) returns -\$2594.66

For the same loan, if payments are due at the beginning of the period, the payment is:

pmt(0.1/12, 8, 20000, 0, 1) returns -\$2573.21

pv(*rate*, *nper*, *pmt* [, *fv* [, *type*]])

Returns the present value of an investment. The result has a default number format of money. The present value is the total amount that a series of future payments is worth now. For example, when you borrow money, the loan amount is the present value to the lender.

Rate is the interest rate per period.

Nper is the total number of payment periods in an annuity.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *fv* argument.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, you must include the *pmt* argument.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Example:

What is the present value of \$100 paid to you at the end of each month during a five-year period? The interest rate is 7%.

pv(0.07/12, 12*5, 100, 0, 0) returns -\$5050.20

rate(*nper*, *pmt*, *pv* [, *fv* [, *type*]])

Returns the interest rate per period of an annuity.

Nper is the total number of payment periods in an annuity.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *fv* argument.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, it is assumed to be 0.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Rate is calculated by iteration and can have zero or more solutions. If the successive results of **rate** do not converge, a NULL value will be returned.

Example:

What is the interest rate of a five-year \$10,000 annuity loan with monthly payments of \$300?

rate(60,-300,10000) returns 2.18%

This is the monthly rate, which has to be multiplied with 12 in order to get the annual rate, in this case 26.1%.

Mathematical Constants and Parameter-Free Functions

These functions do not have any parameters. The parentheses are however still needed.

e()

Base of the natural logarithms, e. The function returns 2.71828...

false()

Returns a dual value with text value 'false' and numeric value 0, which can be used as logical false in expressions.

pi()

The function returns 3.14159...

rand()

Returns a random number between 0 and 1.

true()

Returns a dual value with text value 'true' and numeric value -1, which can be used as logical true in expressions.

Counter Functions

These functions do not have any parameters. The parentheses are however still needed.

RecNo()

Returns an integer for the number of the currently read row of the input table. The first record is number 1.

RowNo()

Returns an integer for the position of the current row in the resulting QlikView input table. In contrast to **RecNo()**, which counts the records in the raw data table, the **RowNo()** function doesn't count records excluded by **where** clauses and is not reset when a raw data table is concatenated to another. The first row is number 1.

Example:

Tab1.csv (raw data) Tab2.csv (raw data)

A	B
1	aa
2	cc
3	ee

A	C
5	xx
4	yy
6	zz

QVTab:

```
Load *, RecNo(), RowNo() from Tab1.csv
where A<>2;
Load *, RecNo(), RowNo() from Tab2.csv
where A<>5;
```

QVTab (QV input table)

A	B	RECNO()	ROWNO()
1	aa	1	1
3	cc	3	2
4	yy	2	3
6	zz	3	4

IterNo()

This function only has a meaning if used together with a **while** clause (see the documentation on “Load” on page 309). **IterNo()** returns an integer indicating for which time one single record is evaluated in a **load** statement with a **while** clause. The first iteration has number 1.

Example:

Tab1.csv

A	B	C
1	5	xx
2	4	yy

```
Load *, IterNo() as D from Tab1.csv  
while IterNo() <=3;
```

QVTab

A	B	C	D
1	5	xx	1
1	5	xx	2
1	5	xx	3
2	4	yy	1
2	4	yy	2
2	4	yy	3

autonumber(*expression* [, *AutoID*])

Returns a unique integer value for each distinct evaluated value of *expression* encountered during the script execution. This function can be used e.g. for creating a compact memory representation of a complex key.

In order to create multiple counter instances if the **autonumber** function is used on different keys within the script, an optional parameter *AutoID* can be used to name each counter.

Examples:

autonumber(Region&Year&Month)

autonumber(Region&Year&Month, 'Ctr1')

autonumberhash128(*expression* {, *expression*})

Calculates a 128-bit hash of the combined input expression values and returns a unique integer value for each distinct hash value encountered during the script execution. This function can be used e.g. for creating a compact memory representation of a complex key.

Example:

autonumberhash128 (Region, Year, Month)

autonumberhash256(*expression* {, *expression*})

Calculates a 256-bit hash of the combined input expression values and returns a unique integer value for each distinct hash value encountered during the script execution. This function can be used e.g. for creating a compact memory representation of a complex key.

Example:

autonumberhash256 (Region, Year, Month)

fieldvaluecount (*fieldname*)

Returns the number of distinct values in a field. *Fieldname* must be given as a string (e.g. a quoted literal).

Example:

let x =fieldvaluecount ('Alfa');

String Functions

ord(*s*) ASCII number of first character of string *s*. The result is an integer.

Example:

ord('A') returns the number 65.

chr(*n*) The ASCII character corresponding to number *n*. The result is a string.

Example:

chr(65) returns the string 'A'.

len(*s*) Length of string *s*. The result is an integer.

Example:

len(Name) where Name = 'Peter' returns 5.

left(*s*, *n*)

Substring of the string *s*. The result is a string consisting of the first *n* characters of *s*.

Examples:

left('abcdef',3) returns 'abc'.

left(Date, 4) where Date = 1997-07-14 returns 1997.

For a more complicated example, see the **index** function below.

right(*s* , *n*)

Substring of the string *s*. The result is a string consisting of the last *n* characters of *s*.

Examples:

right('abcdef',3) returns 'def'.

right(Date,2) where Date = 1997-07-14 returns 14.

mid(*s* , *n1* [, *n2*])

Substring of the string *s*. The result is the string starting at character *n1* with the length of *n2* characters. If *n2* is omitted, the function returns the rightmost part of the string starting at character *n1*. The positions in the string are numbered from 1 and up.

Examples:

mid('abcdef',3) returns 'cdef'.

mid('abcdef',3, 2) returns 'cd'.

mid(Date,3) where Date = 970714 returns 0714

mid(Date,3,2) where Date = 970714 returns 07

For a more complicated example, see the **index** function below.

index(*s1* , *s2* [, *n*])

Position of a substring. This function gives the starting position of the *n*th occurrence of substring *s2* in string *s1*. If *n* is omitted, the first occurrence is assumed. If *n* is negative, the search is made starting from the end of string *s1*. The result is an integer. The positions in the string are numbered from 1 and up.

Examples:

index('abcdefg', 'cd') returns 3

index('abcdabcd', 'b', 2) returns 6

index('abcdabcd', 'b', -2) returns 2

left(Date, index(Date, '-') - 1)

where Date = 1997-07-14 returns 1997

mid(Date, index(Date, '-' , 2) - 2, 2)

where Date = 1997-07-14 returns 07

upper(*text expression*)

Forces upper case for all the data in the expression.

Example:

upper('abcd') returns 'ABCD'.

lower(*text expression*)

Forces lower case for all the data in the expression.

Example:

lower('abcD') returns 'abcd'.

capitalize(*s*)

Returns the string *s* with all words capitalized.

Examples:

capitalize('my little pony') returns 'My Little Pony'.

capitalize('AA bb cC Dd') returns 'Aa Bb Cc Dd'.

repeat(*s*, *n*)

Forms a string consisting of the string *s* repeated *n* times.

Example:

repeat('*', rating) where rating = 4 returns '****'.

ltrim(*s*)

Returns the string *s* trimmed of any leading spaces.

Examples:

ltrim(' abc') returns 'abc'.

ltrim('abc ') returns 'abc '.

rtrim(*s*)

Returns the string *s* trimmed of any trailing spaces.

Examples:

rtrim(' abc') returns ' abc'.

rtrim('abc ') returns 'abc'.

trim(*s*)

Returns the string *s* trimmed of any leading and trailing spaces.

Examples:

trim(' abc') returns 'abc'.

trim('abc ') returns 'abc'.

trim(' abc ') returns 'abc'.

KeepChar(*s1*, *s2*)

Returns the string *s1* less all characters not contained in string *s2*.

Example:

keepchar('a1b2c3' , '123') returns '123'.

PurgeChar(*s1* , *s2*)

Returns the string *s1* less all characters contained in string *s2*.

Example:

purgechar('a1b2c3' , '123') returns 'abc'.

replace (*string* , *fromstring* , *tostring*)

Returns a *string* after replacing all occurrences of a given substring within the string with another substring. The function is non-recursive and works from left to right.

String is the original string.

fromstring is a string which may occur one or more times within *string*.

tostring is the string which will replace all occurrences of *fromstring* within *string*.

Example:

replace('abcde','cc','xyz')

TextBetween (*s* , *beforetext* , *aftertext* [, *n*])

Returns the text between the *n*th occurrence of *beforetext* and the immediately following occurrence of *aftertext* within the string *s*.

Examples:

TextBetween(<abc> , '<' , '>') returns 'abc'

TextBetween(<abc><de> , '<' , '>',2) returns 'de'

subfield(*s* , '*delimiter*' [, *index*])

In its three parameter version, this script function returns a given substring from a larger string (*s*) with delimiters ('*delimiter*'). *Index* is an optional integer denoting which of the substrings should be returned. If *index* is omitted when **subfield** is used in a field expression in a **load** statement, the **subfield** function will cause the **load** statement to automatically generate one full record of input data for each substring that can be found in *s*.

In its two parameter version, the **subfield** function generates one record for each substring that can be taken from a larger string (*s*) with delimiters ('*delimiter*'). If several **subfield** functions are used

in the same **load** statement, the cartesian product of all combinations will be generated.

The returned strings are not trimmed from leading and trailing spaces.

Example 1 (three parameters):

subfield(S, ';',2) returns 'cde' if S is 'abc;cde:efg'

subfield(S, ' ',1) returns NULL if S is an empty string

subfield(S, ' ',1) returns an empty string if S is ' '

Example 2 (two parameters):

MEETINGID	START	STOP	PARTICIPANTS
Dev	9:00	10:00	John;Peter;Ann
Market	11:30	12:45	Jean;Julia;Robert;Peter

MyTab:

```
Load MeetingID, Start, Stop,
subfield(Participants, ';') as People
from Tab1.csv;
```

MEETINGID	START	STOP	PEOPLE
Dev	9:00	10:00	John
Dev	9:00	10:00	Peter
Dev	9:00	10:00	Ann
Market	11:30	12:45	Jean
Market	11:30	12:45	Julia
Market	11:30	12:45	Robert
Market	11:30	12:45	Peter

Example 3 (two parameters):

BATCH	PRODUCTS	MACHINES
A	Alpha,Beta	Slicer;Cutter
B	Gamma	Slicer;Packer;Loader

```
MyTab:
Load Batch,
subfield(Products, ',')
as Products,
subfield(Machines, ';')
as Machines
from Tab2.csv;
```

BATCH	PRODUCTS	MACHINES
A	Alpha	Slicer
A	Alpha	Cutter
A	Beta	Slicer
A	Beta	Cutter
B	Gamma	Slicer
B	Gamma	Packer
B	Gamma	Loader

evaluate(s)

If the text *s* can be interpreted as a valid QlikView expression, the evaluated result of the expression will be returned. If *s* is not a valid expression, NULL is returned.

Example:

evaluate(5*8) returns 40

findoneof (text, characterset [, n])

returns the position of the *n*th occurrence in the string *text* of any of the characters found in the string *characterset*. If *n* is omitted the position of the first occurrence is returned. If there is no match 0 is returned.

Examples:

findoneof('my example text string', 'et%s') returns 4

findoneof('my example text string', 'et%s', 3) returns 12

findoneof('my example text string', 'æ%s&') returns 0'

substringcount(*text* , *substring*)

Returns the number of times the string substring appears within the string text. The result is an integer. If there is no match, 0 is returned.

Example:

substringcount ('abcdefgdcxyz' , 'cd') returns 2

applycodepage(*codepage*,*text*)

Applies a different codepage to the field or text stated in the expression. The codepage must be in number format.

hash128(*expression* { , *expression* })

Returns a 128-bit hash of the combined input expression values. The result is a string.

Example:

hash128(*Region*, *Year*, *Month*)

hash160(*expression* { , *expression* })

Returns a 160-bit hash of the combined input expression values. The result is a string.

Example:

hash160(*Region*, *Year*, *Month*)

hash256(*expression* { , *expression* })

Returns a 256-bit hash of the combined input expression values. The result is a string.

Example:

hash256(*Region*, *Year*, *Month*)

Mapping Functions

ApplyMap('*mapname*' , *expr* [, *defaultexpr*])

Maps any expression on a previously loaded mapping table. *Map-name* is the name of a mapping table previously loaded by a **mapping load** or **mapping select** statement (see page 318). The name must be quoted with single quotes. *Expr* is the expression whose result shall be mapped. *Defaultexpr* is an optional expression, which will be used as default mapping value if the mapping table does not contain any matching value for *expr*. If no default is provided, the value of *expr* is returned as is.

Examples:

```
// Assume the following mapping table
map1:
mapping load * inline [
x,y
1,one
2,two
3,three];
```

ApplyMap('map1' , MyVal) returns 'two' if MyVal = 2

ApplyMap('map1' , MyVal) returns 4 if MyVal = 4

ApplyMap('map1' , MyVal , 'x')
returns 'x' if MyVal <> 1, 2 or 3

ApplyMap('map1' , MyVal , null())
returns null if MyVal <> 1, 2 or 3

ApplyMap('map1' , MyVal , null())
returns 'one' if MyVal = 1

MapSubstring('mapname' , expr)

This function can be used for mapping parts of any expression on a previously loaded mapping table. The mapping is case sensitive and non-recursive. The substrings are mapped from the left to the right. *Mapname* is the name of a mapping table previously read by a **mapping load** or a **mapping select** statement. The name must be enclosed by single straight quotation marks. *Expr* is the expression whose result should be mapped by substrings.

Examples:

```
// Assume the following mapping table
map1:
mapping [
x,y
1,<one>
aa,XYZ
x,b];
```

MapSubstring('map1' , 'A123') returns 'A<one>23'

MapSubstring('map1' , 'baaar') returns 'bXYZar'

MapSubstring('map1' , 'xaa1') returns 'bXYZ<one>'

Inter-Record Functions

These functions are used when a value from previously loaded records of data is needed for the evaluation of the current record.

previous(*expression*)

Returns the value of *expression* using data from the previous input record that was not discarded due to a **where** clause. In the first record of an input table the function will return NULL. The **previous** function may be nested in order to access records further back. Data is fetched directly from the input source, which makes it possible to refer also to fields which have not been loaded into QlikView and stored in its internal database.

Examples:

```
Load *, Sales / previous(Sales) as Increase
from..;
```

```
Load A, previous(previous(A)) as B from..;
```

peek(*fieldname* [, *row* [, *tablename*]])

Returns the contents of the field *fieldname* in the record specified by *row* in the input table *tablename*. Data is fetched from the internal QlikView database.

Fieldname must be given as a string (e.g. a quoted literal).

Row must be an integer. 0 denotes the first record, 1 the second record and so on. Negative numbers indicate order from the end of the table. -1 denotes the last record read.

If no row is stated, -1 is assumed.

Tablename is a table label without the ending colon. If no tablename is stated, the current table is assumed. If the **peek** function is used outside the load statement or referring to another table, this parameter must be included.

Examples:**peek**('Sales')

returns the value of Sales in the previous record read (equivalent to **previous**(Sales)).

peek('Sales', 2)

returns the value of Sales from the third record read from the current input table.

peek('Sales', -2)

returns the value of Sales from the second last record read into the current input table.

peek('Sales',0,'Tab1')

returns the value of Sales from the first record read into the input table labeled Tab1.

Load A, B, numsum(B, peek('Bsum')) as Bsum...;
creates an accumulation of B in Bsum.

exists(*field [, expr]*)

Determines whether a specific field value exists in a specified field of the data loaded so far. *Field* is a name or a string expression evaluating to a field name. The field must exist in the data loaded so far by the script. *Expr* is an expression evaluating to the field value to look for in the specified field. If omitted, the current record's value in the specified field will be assumed.

Examples:

exists(Month, 'Jan')

returns -1 (true) if the field value *Jan* can be found in the current contents of the field *Month*.

exists(IDnr, IDnr)

returns -1 (true) if the value of the field *IDnr* in the current record already exists in any previously read record containing that field.

exists(IDnr)

is identical with the previous example.

Examples:

```
Load Employee, ID, Salary from
Employees.csv;
Load FirstName&' '&LastName as Employee,
Comment
from Citizens.csv
where exists (Employee, FirstName&'
'&LastName);
```

Only comments regarding citizens who are employees are read.

```
Load A, B, C from Employees.csv
where not exists(A);
```

This is equivalent to performing **distinct load** on field A.

fieldvalue(*fieldname, n*)

Returns the field value found in position *n* of the field *fieldname* (by load order). *Fieldname* must be given as a string value, e.g. the field name must be enclosed by single quotes. The first field value is returned for *n*=1. If *n* is larger than the number of field values, NULL is returned.

Example:

fieldvalue('Helptext', 5)

fieldindex(*fieldname*, *value*)

Returns the position of the field value *value* in the field *fieldname* (by load order). If *value* cannot be found among the field values of the field *fieldname*, 0 is returned. *Fieldname* must be given as a string value, e.g. the field name must be enclosed by single quotes.

Example:

fieldindex('Name', 'John Doe')

lookup(*fieldname*, *matchfieldname*, *matchfieldvalue* [, *tablename*])

Returns the value of *fieldname* corresponding to the first occurrence of the value *matchfieldvalue* in the field *matchfieldname*.

Fieldname, *matchfieldname* and *tablename* must be given as strings (e.g. quoted literals).

The search order is the load order unless the table is the result of complex operations such as joins, in which case the order is not well defined.

Both *fieldname* and *matchfieldname* must be fields in the same table, specified by *tablename*. If *tablename* is omitted the current table is assumed.

If no match is found, null is returned.

Example:

lookup('Price', 'ProductID', 'InvoicedProd', 'pricelist')

Conditional Functions

if(*condition*, *then* [, *else*])

The three parameters *condition*, *then* and *else* are all expressions. The first one, *condition*, is interpreted logically. The two other ones, *then* and *else*, can be of any type. They should preferably be of the same type. The third parameter, *else*, is optional. If excluded it will be assumed null. If *condition* is true, the function returns the value of the expression *then*. If *condition* is false, the function returns the value of the expression *else*.

Example:

if(Amount >= 0, 'OK', 'Alarm')

alt(*case1* [, *case2* , *case3* , ...] , *otherwise*)

The **alt** function returns the first of the parameters that has a valid number representation. If no such match is found, the last parameter will be returned. Any number of parameters can be used.

Example:

```
alt( date#( dat,'YYYY/MM/DD' ) ,  
date#( dat,'MM/DD/YYYY' ) ,  
date#( dat,'MM/DD/YY' ) ,  
'No valid date' )
```

will test if the variable *dat* contains a date according to any of the three specified date formats. If so, it will return the original string and a valid number representation of a date. If no match is found, the text 'No valid date' will be returned (without any valid number representation).

pick(*n* , *expr1* [, *expr2* , ... , *exprN*])

Returns the *n*th expression in the list. *n* is an integer between 1 and *N*.

Examples:

```
pick( 2 , 'A' , 'B' , 4 ) returns 'B'  
pick( 3 , 'A' , 'B' , 4 ) returns 4
```

match(*s* , *expr1* [, *expr2* , ... , *exprN*])

Compares the string *s* to a list of strings or string expressions. The result of the comparison is an integer indicating which of the comparison strings/expressions matched. If no match is found, 0 is returned. The match function performs a case sensitive comparison.

Examples:

```
match( M , 'Jan' , 'Feb' , 'Mar' )  
  where M is 'Feb' returns 2  
match( M , 'Jan' , 'Feb' , 'Mar' )  
  where M is 'Apr' or 'jan' returns 0
```

mixmatch(*s* , *expr1* [, *expr2* , ... , *exprN*])

Compares the string *s* to a list of strings or string expressions. The result of the comparison is an integer indicating which of the comparison strings/expressions matched. If no match is found, 0 is returned. The mixmatch function performs a case insensitive comparison.

Example:

mixmatch(M , 'Jan' , 'Feb' , 'Mar')
 where M is 'jan' returns 1

wildmatch(*s* , *expr1* [, *expr2* , ...*exprN*])

Compares the string *s* to a list of strings or string expressions. The result of the comparison is an integer indicating which of the comparison strings/expressions that matched. If no match is found, 0 is returned. The wildmatch function performs a case insensitive comparison. The wildcard characters * and ? are allowed in the comparison string.

Examples:

wildmatch(M , 'ja*' , 'fe?' , 'mar')
 where M is 'January' returns 1

wildmatch(M , 'ja*' , 'fe?' , 'mar')
 where M is 'fex' returns 2

Wildmatch5(*s*,*expr1* [,*expr2*,...*exprN*])

Works as the above wildmatch function, but is used only for compatibility reasons with QlikView version 5.

class(*expression* , *interval* [, *label* [, *offset*]])

Creates a classification of *expressions*. The bin width is determined by the number set as *interval*. The result is shown as $a \leq x < b$, where *a* and *b* are the upper and lower limits of the bin. The *x* can be replaced by an arbitrary string stated in *label*. 0 is normally the default starting point of the classification. This can be changed by adding an *offset*.

Examples:

class(var,10) with var = 23 returns '20<=x<30'

class(var,5,'value') with var = 23 returns '20<= value <25'

class(var,10,'x',5) with var = 23 returns '15<=x<25'

Logical Functions

IsNum(*expr*)

Returns -1 (true) if the expression can be interpreted as a number, otherwise 0 (false).

IsPartialReload()

Returns -1 (true) if the current reload is partial, otherwise 0 (false).

IsText(*expr*)

Returns -1 (true) if the expression has a text representation, otherwise 0 (false).

Null Functions

Null() Returns a real NULL value.

IsNull(*expr*)

Returns -1 (true) if *expr* returns NULL, otherwise 0 (false).

System Functions

OSuser()

Returns a string containing the name of the current user as returned by the operating system.

QVuser()

Returns a string containing the name of the current QlikView user as defined in a **section access**.

qlikviewversion()

Returns the full QlikView version and build number (e.g. 7.52.3797.0409.3) as a string. This function is only available in QlikView builds after 7.52.3795.

DocumentName()

Returns a string containing the name of the QlikView document currently being read without path but with extension.

Example:

```
Load *, documentname() as X from  
c:\UserFiles\Mydoc.qvw
```

will return 'MyDoc.qvw' in field X in each record.

DocumentPath()

Returns a string containing the full path to the QlikView document currently being read.

Example:

```
Load *, documentpath() as X from  
c:\UserFiles\Mydoc.qvw
```

will return 'C:\UserFiles\MyDoc.qvw' in field X in each record.

DocumentTitle()

Returns a string containing the title of the QlikView document currently being read.

Example:

```
Load *, documenttitle() as X from
c:\UserFiles\Mydoc.qvw
```

will return 'My Document' in field X in each record.

ComputerName()

Returns a string containing the name of the computer as returned by the operating system.

ReloadTime()

Returns a timestamp for the last finished execution of the script.

GetRegistryString(*path*, *key*)

Returns the value of a named registry *key* with a given registry *path*.

Example:

```
getregistrystring('HKEY_LOCAL_MACHINE\SOFTWARE\QlikTech\QlikViewServer\Settings 7','EnableSessionLog')
```

Input(*str cue* [, *str caption*])

This function can be used in the script only and opens an input box that prompts the user for a value during the script execution. The parameters *cue* and *caption* are used as message and caption texts, respectively. The function returns the entered value.

The input box function returns NULL if the dialog is cancelled, closed or cannot be shown.

Example:

```
Load
    Input('Enter value', 'Input box') as v,
    Recno () as r
autogenerate 3;
```

MsgBox(*str msg* [, *str caption*] [, *mb_buttons*] [, *mb_icons*] [, *mb_defbutton*]]]])

This function can be used in the script only and opens a message box during the script execution. The parameters *msg* and *caption* are used as message and caption texts, respectively. The parameter *mb_buttons* defines what buttons will be shown in the message box, according to:

0 or 'OK' for a single OK button,
1 or 'OKCANCEL' for two buttons, OK and Cancel,
2 or 'ABORTRETRYIGNORE' for three buttons, Abort, Retry and Ignore,
3 or 'YESNOCANCEL' for three buttons, Yes, No and Cancel,
4 or 'YESNO' for two buttons, Yes and No,
5 or 'RETRYCANCEL' for two buttons, Retry and Cancel.

The parameter *mb_icons* defines what icon will be shown in the message box, according to:

0 or empty string for no icon,
16 or 'ICONHAND' for an icon with an X, used for critical errors,
32 or 'ICONQUESTION' for an icon with a question mark,
48 or 'ICONEXCLAMATION' for icon with an exclamation mark, used for minor errors, cautions and warnings
64 or 'ICONASTERISK' icon with an “i”, used for informational messages.

The parameter *mb_defbutton* defines what button will have focus when the message box is shown, according to:

0 or 'DEFBUTTON1' if the first button should have focus,
256 or 'DEFBUTTON2' if the second button should have focus,
512 or 'DEFBUTTON3' if the third button should have focus,
768 or 'DEFBUTTON4' if the fourth button should have focus.

The function returns an integer that shows what button has been pressed by the user, according to:

1 for OK,
2 for Cancel,
3 for Abort,
4 for Retry,
5 for Ignore,
6 for Yes,
7 for No

The parameter 3, 4 and 5 will internally be added, so if numeric values other than the above mentioned ones are used, you may get an unexpected combination of icons and buttons.

The message box function returns NULL if the dialog cannot be shown.

Example:

```
Load
MsgBox('Message 2', 'msgbox', 'OKCANCEL',
'ICONASTERISK') as x,
2 as r
autogenerate 1;
```

File Functions**Attribute(filename, attributename)**

Returns the value of the meta tags of different file formats, e.g. MP3, WMA, WMV and JPG files, as text.

Filename is the name of a media file including path, if needed.

Attributename is the name of a meta tag.

If the file *filename* does not exist, is not a supported file format or does not contain a meta tag named *attributename*, null will be returned.

A large number of meta tags can be used, e.g. 'Artist' or 'Date Picture Taken'. The supported tags can automatically be generated in the script. The keyboard shortcut for this generation is CTRL + Q,J,P,G for jpg files, CTRL + Q,M,P,3 for mp3 files and CTRL + Q,W,M,A for wma files.

Example:

```
Attribute('File', 'Title') as X,
```

returns the mp3 tag 'title' in field X in each record.

FileBasename()

Returns a string containing the name of the text file currently being read without path and extension.

Example:

```
Load *, filebasename() as X from
C:\UserFiles\abc.txt;
```

will return 'abc' in field X in each record.

FileDir()

Returns a string containing the path to the directory of the text file currently being read.

Example:

```
Load *, filedir() as X from  
C:\UserFiles\abc.txt;
```

will return 'C:\UserFiles' in field X in each record.

FileExtension()

Returns a string containing the extension of the text file currently being read.

Example:

```
Load *, fileextension() as X from  
C:\UserFiles\abc.txt;
```

will return 'txt' in field X in each record.

FileName()

Returns a string containing the name of the text file currently being read without path but with extension.

Example:

```
Load *, filename() as X from  
C:\UserFiles\abc.txt;
```

will return 'abc.txt' in field X in each record.

FilePath()

Returns a string containing the full path to the text file currently being read.

Example:

```
Load *, filepath() as X from  
C:\UserFiles\abc.txt;
```

will return 'C:\UserFiles\abc.txt' in field X in each record.

FileSize([filename])

Returns an integer containing the size in bytes of the file *filename* or if no *filename* is specified of the text file currently being read.

Examples:

filesize('xyz.xls') will return the size of the file xyz.xls.

```
Load *, filesize() as X from abc.txt;
```

will return the size of the specified file (abc.txt) as an integer in field X in each record read.

FileTime([filename])

Returns a timestamp for the date and time of the last modification of the file *filename*. If no *filename* is specified, the function will refer to the currently read text file.

Examples:

filetime('xyz.xls') will return the timestamp of the last modification of the file xyz.xls.

```
Load *, filetime() as X from abc.txt;
```

will return the date and time of the last modification of the file abc.txt as a timestamp in field X in each record read.

ConnectionString()

Returns the active connect string for ODBC or OLEDB connection. Returns an empty string if no **Connect** statement has been executed or after a **Disconnect** statement.

GetFolderPath

Returns the value of the Microsoft Windows SHGetFolderPath function and returns the path for e.g. My Music. Note that the function does not use the spaces seen in Windows Explorer.

Examples:

```
GetFolderPath('MyMusic')
GetFolderPath('MyPictures')
GetFolderPath('MyVideos')
GetFolderPath('MyReceivedFiles')
GetFolderPath('MyShapes')
GetFolderPath('MyMusic')
GetFolderPath('ProgramFiles')
GetFolderPath('Windows')
```

QvdCreateTime(filename)

Returns the XML-header time stamp from a QVD file if any (otherwise NULL).

Filename is name of a QVD file including path, if needed.

Examples:

```
QvdCreateTime( 'MyFile.qvd' )
```

```
QvdCreateTime( 'C:\MyDir\MyFile.qvd' )
```

QvdNoOfRecords(filename)

Returns the number of records currently in a QVD file.

Filename is name of a QVD file including path, if needed.

Examples:

QvdNoOfRecords('MyFile.qvd')

QvdNoOfRecords('C:\MyDir\MyFile.qvd')

QvdNoOfFields(*filename*)

Returns the number of fields in a QVD file.

Filename is name of a QVD file including path, if needed.

Examples:

QvdNoOfFields('MyFile.qvd')

QvdNoOfFields('C:\MyDir\MyFile.qvd')

QvdFieldName(*filename, field_no*)

Returns the name of field number *field_no*, if it exists in a QVD file (otherwise NULL).

Filename is name of a QVD file including path, if needed.

Fieldno is the number of the field (0 based) within the table contained in the QVD file.

Examples:

QvdFieldName('MyFile.qvd' 3)

QvdFieldName('C:\MyDir\MyFile.qvd' 5)

QvdTableName(*filename*)

Returns the name of the table contained in a QVD file.

Filename is name of a QVD file including path, if needed.

Examples:

QvdTableName('MyFile.qvd')

QvdTableName('C:\MyDir\MyFile.qvd')

Table Functions

NoOfRows(*TableName*)

Returns the number of rows (records) in a previously loaded input table. If the function is used within a **load** statement, it must not reference the table currently being loaded.

Examples:

```

tab1:
Load * from abc.csv;
Let a = NoOfRows('tab1');

```

NoOfFields(*TableName*)

Returns the number of fields in a previously loaded input table. If the function is used within a **load** statement, it must not reference the table currently being loaded.

Examples:

```

tab1:
Load * from abc.csv;
Let a = NoOfFields('tab1');

```

FieldNumber(*field*, *TableName*)

Returns the number of a specified field within a previously loaded input table. If the function is used within a **load** statement, it must not reference the table currently being loaded.

Examples:

```

tab1:
Load * from abc.csv;
LET a = FieldNumber('Customer', 'tab1');

```

FieldName(*nr*, *TableName*)

Returns the name of the field with the specified number within a previously loaded input table. If the function is used within a **load** statement, it must not reference the table currently being loaded.

Examples:

```

tab1:
Load * from abc.csv;
LET a = FieldName(4, 'tab1');

```

NoOfTables()

Returns the number of tables previously loaded.

TableName(*table_number*)

Returns the name of the table with the specified number.

TableNumber(*table_name*)

Returns the number of a specified table.

Document Functions

ReportComment(*report_number*)

Returns the comment of the report with the specified number within the active document.

ReportName(*report_number*)

Returns the name of the report with the specified number within the active document.

ReportID(*report_number*)

Returns the id of the report with the specified number within the active document.

ReportNumber(*report_id_or_name*)

Returns the number of the report with the specified id or name within the active document.

NoOfReports()

Returns the number of reports in the active document.

Date and Time Functions

In the following examples, the default date format YYYY-MM-DD (ISO standard) is assumed. See further on page 294 in book II for a description of date and time formats in QlikView.

day(*date*)

Day. The result is a number.

Example:

day(Date) where Date = 1971-10-30 returns 30.

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. To solve this problem, you can either change the settings or use the **date#** interpretation function (see under “Number Interpretation Functions” on page 449). For further information about date and time functions see page 294 in book II.

week(*date*)

Week number. The result is a number.

Example:

week(Date) where Date = 1971-10-30 returns 43.

A week begins on Monday. Week #1 is the first week with four or more days in the new year.

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

month(*date*)

Month. The result is a text string, but can be formatted as a number.

Example:

month(Date) where Date = 1971-10-30 returns Oct.

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

year(*date*)

Year. The result is a number.

Example:

year(Date) where Date = 1971-10-30 returns 1971.

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

weekday(*date*)

Week day. The result is a number between 0-6, where 0 indicates Monday.

Example:

weekday(Date) where Date = 1971-10-30 returns 5.

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

weekyear(*date*)

The year to which the week number belongs. The week number ranges between 1 and approximately 52. Some years week #1 starts in December, e.g. December 1997. Other years start with week #53 of previous year, e.g. January 1999. For those few days when the week number belongs to another year, the functions **year** and **week-year** will return different values.

Examples:

weekyear(Date) where Date = 1996-12-30 returns 1997.

weekyear(Date) where Date = 1997-01-02 returns 1997.

weekyear(Date) where Date = 1997-12-30 returns 1997.

weekyear(Date) where Date = 1999-01-02 returns 1998.

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

MakeDate(YYYY [, MM [, DD]])

Returns a date calculated from the year *YYYY*, the month number *MM* and the day number *DD*. If no month number is stated, 1 (January) is assumed. If no day number is stated, 1 (the 1st) is assumed.

Examples:

(date format YYYY-MM-DD assumed)

makedate(1999) returns 1999-01-01

makedate(99) returns 0099-01-01

makedate(1992, 12) returns 1992-12-01

makedate(1999, 2, 14) returns 1999-02-14

MakeWeekDate(YYYY, WW [, D])

Returns a date calculated from the year *YYYY*, the week number *WW* and the day-of-week *D*, where 0 denotes Monday and 6 denotes Sunday. Week number 1 is the first week containing four days or more of the year. If no day-of-week is stated, 0 (Monday) is assumed.

Examples:

(date format YYYY-MM-DD assumed)

makeweekdate(1999,6,6) returns 1999-02-14

makeweekdate(99,6) returns 0099-02-02

MakeTime(hh [, mm [, ss [, .fff]]])

Returns a time calculated from the hour *hh*, the minute *mm* and the second *ss* with fraction *fff* down to a millisecond. If no minute is stated, 00 is assumed. If no second is stated, 00 is assumed. If no fraction of seconds is stated, .000 is assumed.

Examples:

(time format hh:mm:ss assumed)

maketime(22) returns 22:00:00

maketime(22, 17) returns 22:17:00

maketime(22, 17, 52) returns 22:17:52

hour(*expr*)

Returns an integer representing the hour when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

Examples:

(time format hh:mm:ss assumed)

hour(time) returns 9 when time='09:14:36'

hour(0.5555) returns 13 (0.5555 = 13:19:55)

minute(*expr*)

Returns an integer representing the minute when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

Examples:

(time format hh:mm:ss assumed)

minute(time) returns 14 when time='09:14:36'

minute(0.5555) returns 19 (0.5555 = 13:19:55)

second(*expr*)

Returns an integer representing the second when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

Examples:

(time format hh:mm:ss assumed)

second(time) returns 36 when time='09:14:36'

second(0.5555) returns 55 (0.5555 = 13:19:55)

now([*timer_mode*])

Returns a timestamp from the system clock. The *timer_mode* may have the following values:

- 0 Time at script run
- 1 Time at function call
- 2 Time when the document was opened

Default *timer_mode* is 1. The *timer_mode=1* should be used with caution, since it polls the operating system every second and hence could slow down the system.

today([*timer_mode*])

Returns the date from the system clock. The *timer_mode* may have the following values:

- 0 Date at script run
- 1 Date at function call
- 2 Date when the document was opened

Default *timer_mode* is 2. The *timer_mode=1* should be used with caution, since it polls the operating system every second and hence could slow down the system.

LocalTime([*timezone* [, *ignoreDST*]])

Returns a timestamp of the current time from the system clock for a specified time zone. The *timezone* is specified as a string containing any of the geographical places listed under Time Zone in the Windows Control Panel for Date and Time or as a string in the form 'GMT+hh:mm'. If no *timezone* is specified the local time will be returned. If *ignoreDST* is true daylight savings time will be ignored.

Examples:

```
localtime ('Paris')
localtime ('GMT+01:00')
localtime ('Paris', true)
localtime ()
```

ConvertToLocalTime(*timestamp* [, *place* [, *ignore_dst=false*]]])

Converts a UTC or GMT timestamp to local time as a dual value. The place can be any of a number of cities, places and time zones around the world, e.g. Abu Dhabi, Adelaide, Alaska, Almaty, Amsterdam, Arizona, Astana, Athens, Atlantic Time (Canada), Auckland, Azores, Baghdad, Baku, Bangkok, Beijing, Belgrade, Berlin, Bern, Bogota, Brasilia, Bratislava, Brisbane, Brussels, Bucharest, Budapest, Buenos Aires, Cairo, Canberra, Cape Verde Is., Caracas, Casablanca, Central America, Central Time (US & Canada), Chennai, Chihuahua, Chongqing, Copenhagen, Darwin, Dhaka, Eastern Time (US & Canada), Edinburgh, Ekaterinburg, Fiji, Georgetown, Greenland, Greenwich Mean Time : Dublin, Guadalajara, Guam, Hanoi, Harare, Hawaii, Helsinki, Hobart, Hong Kong, Indiana (East), International Date Line West, Irkutsk, Islamabad, Istanbul, Jakarta, Jerusalem, Kabul, Kamchatka, Karachi, Kathmandu, Kolkata, Krasnoyarsk, Kuala Lumpur, Kuwait, Kyiv, La Paz, Lima, Lisbon, Ljubljana, London, Madrid, Magadan, Marshall Is., Mazatlan, Melbourne, Mexico City, Mid-Atlantic, Midway

Island, Minsk, Monrovia, Monterrey, Moscow, Mountain Time (US & Canada), Mumbai, Muscat, Nairobi, New Caledonia, New Delhi, Newfoundland, Novosibirsk, Nuku'alofa, Nuku'alofa, Osaka, Pacific Time (US & Canada), Paris, Perth, Port Moresby, Prague, Pretoria, Quito, Rangoon, Riga, Riyadh, Rome, Samoa, Santiago, Sapporo, Sarajevo, Saskatchewan, Seoul, Singapore, Skopje, Sofia, Solomon Is., Sri Jayawardenepura, St. Petersburg, Stockholm, Sydney, Taipei, Tallinn, Tashkent, Tbilisi, Tehran, Tijuana, Tokyo, Ulaan Bataar, Urumqi, Warsaw, Wellington, West Central Africa, Vienna, Vilnius, Vladivostok, Volgograd, Yakutsk, Yerevan or Zagreb. Also GMT, GMT-01:00, GMT+04:00 etc. are valid places.

The resulting time is adjusted for daylight savings time, unless the third parameter is set to 1 or true().

Examples:

ConvertToLocalTime('2007-11-10 23:59:00', 'Paris') returns '2007-11-11 00:59:00' and the corresponding internal timestamp representation.

ConvertToLocalTime(UTC(), 'GMT-05:00') returns the time for the North American east coast, e.g. New York.

YearToDate(*date* [, *yearoffset* [, *firstmonth* [, *todaydate*]]])

Returns true if *date* falls within the year to date, else false. If none of the optional parameters are used, the year to date means any date within one calendar year from January 1 up to and including the date of the last script execution.

By specifying a *yearoffset* (0 if omitted), the function can be transposed to return true for the same period in another year. Negative *yearoffset* indicates previous years while a positive value indicates coming years. Last year to date is achieved by specifying *yearoffset*=-1.

By specifying a *firstmonth* between 1 and 12 (1 if omitted), the beginning of the year may be moved forward to the first day of any month. If you e.g. want to work with a fiscal year starting May 1, you may specify *firstmonth* = 5.

By specifying a *todaydate* (time of last script execution if omitted), you may move the day used as the upper boundary of the period.

Examples:

assume last reload time = 1999-11-18

yeartodate('1998-11-18') returns false

yeartodate('1999-02-01') returns true
yeartodate('1999-11-18') returns true
yeartodate('1999-11-19') returns false
yeartodate('1998-11-18', -1) returns true
yeartodate('1999-11-18', -1) returns false
yeartodate('1999-04-30', 0, 5) returns false
yeartodate('1999-05-01', 0, 5) returns true

DayNumberOfYear(*date*[,*firstmonth*])

Returns the day number of the year according to a timestamp with the first millisecond of the first day of the year containing *date*.

The function always uses years based on 366 days.

By specifying a *firstmonth* between 1 and 12 (1 if omitted), the beginning of the year may be moved forward to the first day of any month. If you e.g. want to work with a fiscal year starting March 1, you may specify *firstmonth* = 3.

Examples:

DayNumberOfYear(*date*) returns the day number counted from the first of the year.

DayNumberOfYear(*date*,3) returns the number of the day as counted from the first of March.

DayNumberOfQuarter(*date*[,*firstmonth*])

Returns the day number of the quarter according to a timestamp with the first millisecond of the first day of the quarter containing *date*.

The function always uses years based on 366 days.

By specifying a *firstmonth* between 1 and 12 (1 if omitted), the beginning of the year may be moved forward to the first day of any month. If you e.g. want to work with a fiscal year starting March 1, you may specify *firstmonth* = 3.

Examples:

DayNumberOfQuarter(*Date*) returns the day number of the quarter counted from the first day of the first quarter.

DayNumberOfQuarter(*Date*,3) returns the day number of the quarter counted from the first of March

AddMonths(*startdate*, *n* [, *mode*])

Returns the date occurring *n* months after *startdate* or, if *n* is negative, the date occurring *n* months before *startdate*.

By specifying a *mode* (0 if omitted) the date is set to either the unmodified day of the specified month (*mode*=0) or the calculated day as derived from the end of the month (*mode*=1).

Examples:

addmonths ('2003-01-29',3) returns '2003-04-29'
addmonths ('2003-01-29',3,0) returns '2003-04-29'
addmonths ('2003-01-29',3,1) returns '2003-04-28'
addmonths ('2003-01-29',1,0) returns '2003-02-28'
addmonths ('2003-01-29',1,1) returns '2003-02-26'
addmonths ('2003-02-28',1,0) returns '2003-03-28'
addmonths ('2003-02-28',1,1) returns '2003-03-31'

YearStart(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the first millisecond of the first date of the year containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the year which contains *date*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Example:

yearstart ('2001-10-19') returns '2001-01-01' with an underlying numeric value corresponding to '2001-01-01 00:00:00.000'
yearstart ('2001-10-19', -1) returns '2000-01-01' with an underlying numeric value corresponding to '2000-01-01 00:00:00.000'
yearstart ('2001-10-19', 0, 4) returns '2001-04-01' with an underlying numeric value corresponding to '2001-04-01 00:00:00.000'

YearEnd(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the last millisecond of the last date of the year containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the year which contains *date*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_of_year*.

Example:

yearend ('2001-10-19') returns '2001-12-31' with an underlying numeric value corresponding to '2001-12-31 23:59:59.999'

`yearend ('2001-10-19', -1)` returns '2000-12-31' with an underlying numeric value corresponding to '2000-12-31 23:59:59.999'
`yearend ('2001-10-19', 0, 4)` returns '2002-03-31' with an underlying numeric value corresponding to '2002-03-31 23:59:59.999'

YearName(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a four-digit year as display value with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the year containing *date*. *Shift* is an integer, where the value 0 indicates the year which contains *date*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*. The display value will then be a string showing two years.

Examples:

`yearname ('2001-10-19')` returns '2001' with an underlying numeric value corresponding to '2001-01-01 00:00:00.000'

`yearname ('2001-10-19', -1)` returns '2000' with an underlying numeric value corresponding to '2000-01-01 00:00:00.000'

`yearname ('2001-10-19', 0, 4)` returns '2001-2002' with an underlying numeric value corresponding to '2001-04-01 00:00:00.000'

QuarterStart(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the first millisecond of the quarter containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the quarter which contains *date*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

`quarterstart ('2005-10-29')` returns '2005-10-01' with an underlying numeric value corresponding to '2005-10-01 00:00:00.000'

`quarterstart ('2005-10-29', -1)` returns '2005-07-01' with an underlying numeric value corresponding to '2005-07-01 00:00:00.000'

`quarterstart ('2005-10-29', 0, 3)` returns '2005-09-01' with an underlying numeric value corresponding to '2005-09-01 00:00:00.000'

QuarterEnd(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the last millisecond of the quarter containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the quarter which contains *date*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

quarterend ('2005-10-29') returns '2005-12-31' with an underlying numeric value corresponding to '2005-12-31 23:59:59.999'

quarterend('2005-10-29', -1) returns '2005-09-30' with an underlying numeric value corresponding to '2005-09-30 23:59:59.999'

quarterend ('2005-10-29', 0, 3) returns '2005-11-30' with an underlying numeric value corresponding to '2005-11-30 23:59:59.999'

QuarterName(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a display value showing the months of the quarter (formatted according to the *MonthNames* script variable) and year with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the quarter. *Shift* is an integer, where the value 0 indicates the quarter which contains *date*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

quartername ('2005-10-29') returns 'Oct-Dec 2005' with an underlying numeric value corresponding to '2005-10-01 00:00:00.000'

quartername ('2005-10-29', -1) returns 'Jul-Sep 2005' with an underlying numeric value corresponding to '2005-07-01 00:00:00.000'

quartername ('2005-10-29', 0, 3) returns 'Sep-Nov 2005' with an underlying numeric value corresponding to '2005-09-01 00:00:00.000'

MonthStart(*date* [, *shift* = 0])

Returns a value corresponding to a timestamp with the first millisecond of the first date of the month containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the month which contains *date*. Nega-

tive values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

monthstart ('2001-10-19') returns '2001-10-01' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

monthstart ('2001-10-19', -1) returns '2001-09-01' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

MonthEnd(*date* [, *shift* = 0])

Returns a value corresponding to a timestamp with the last millisecond of the last date of the month containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the month which contains *date*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

monthend ('2001-02-19') returns '2001-02-28' with an underlying numeric value corresponding to '2001-02-28 23:59:59.999'

monthend ('2001-02-19', -1) returns '2001-01-31' with an underlying numeric value corresponding to '2001-01-31 23:59:59.999'

MonthName(*date* [, *shift* = 0])

Returns a display value showing the month (formatted according to the *MonthNames* script variable) and year with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the month. *Shift* is an integer, where the value 0 indicates the month which contains *date*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

monthname ('2001-10-19') returns 'Oct 2001' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

monthname ('2001-10-19', -1) returns 'Sep 2001' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

MonthsStart(*n*, *date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the first millisecond of the *n* month period (starting from January 1st) containing *date*. The default output format will be the *DateFormat* set in the script. *N* must be (1), 2, (3), 4 or 6. *Shift* is an integer, where the value 0 indicates the period which contains *date*. Negative values in

shift indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

monthsstart (4, '2001-10-19') returns '2001-09-01' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

monthsstart (4, '2001-10-19', -1) returns '2001-05-01' with an underlying numeric value corresponding to '2001-05-01 00:00:00.000'

monthsstart (4, '2001-10-19', 0, 2) returns '2001-10-01' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

MonthsEnd(*n*, *date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the last millisecond of the *n* month period (starting from January 1st) containing *date*. The default output format will be the *DateFormat* set in the script. *N* must be (1), 2, (3), 4 or 6. *Shift* is an integer, where the value 0 indicates the period which contains *date*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

monthsend (4, '2001-07-19') returns '2001-08-31' with an underlying numeric value corresponding to '2001-08-31 23:59:59.999'

monthsend (4, '2001-10-19', -1) returns '2001-08-31' with an underlying numeric value corresponding to '2001-08-31 23:59:59.999'

monthsend (4, '2001-10-19', 0, 2) returns '2002-01-31' with an underlying numeric value corresponding to '2002-01-31 23:59:59.999'

MonthsName(*n*, *date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a display value showing the months of the period (formatted according to the *MonthNames* script variable) and year with an underlying numeric value corresponding to a timestamp with the first millisecond of the *n* month period (starting from January 1st) containing *date*. *N* must be (1), 2, (3), 4 or 6. *Shift* is an integer, where the value 0 indicates the period which contains *date*. Negative values in *shift* indicate preceding periods and positive values

indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

monthsname (4, '2001-10-19') returns 'Sep-Dec 2001' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

monthsname (4, '2001-10-19', -1) returns 'May-Aug 2001' with an underlying numeric value corresponding to '2001-05-01 00:00:00.000'

monthsname (4, '2001-10-19', 0, 2) returns 'Oct-Jan 2002' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

WeekStart(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a value corresponding to a timestamp with the first millisecond of the first date (Monday) of the calendar week containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the week which contains *date*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

weekstart ('2006-01-12') returns '2006-01-09' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

weekstart ('2006-01-12', -1) returns '2006-01-02' with an underlying numeric value corresponding to '2006-01-02 00:00:00.000'

weekstart ('2006-01-12', 0, 1) returns '2006-01-10' with an underlying numeric value corresponding to '2006-01-10 00:00:00.000'

WeekEnd(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a value corresponding to a timestamp with the last millisecond of the last date (Sunday) of the calendar week containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the week which contains *date*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

weekend ('2006-01-12') returns '2006-01-15' with an underlying numeric value corresponding to '2006-01-15 23:59:59.999'

weekend ('2006-01-12', -1) returns '2006-01-08' with an underlying numeric value corresponding to '2006-01-08 23:59:59.999'

weekend ('2006-01-12', 0, 1) returns '2006-01-16' with an underlying numeric value corresponding to '2006-01-16 23:59:59.999'

WeekName(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a display value showing the year and week number with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the week containing *date*. *Shift* is an integer, where the value 0 indicates the week which contains *date*. Negative values in period indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

weekname ('2006-01-12') returns '2006/02' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

weekname ('2006-01-12', -1) returns '2006/01' with an underlying numeric value corresponding to '2006-01-02 00:00:00.000'

weekname ('2006-01-12', 0, 1) returns '2006/02' with an underlying numeric value corresponding to '2006-01-10 00:00:00.000'

LunarweekStart(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a value corresponding to a timestamp with the first millisecond of the lunar week (consecutive 7 day periods starting on January 1st each year) containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the lunar week which contains *date*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

lunarweekstart ('2006-01-12') returns '2006-01-08' with an underlying numeric value corresponding to '2006-01-08 00:00:00.000'

lunarweekstart ('2006-01-12', -1) returns '2006-01-01' with an underlying numeric value corresponding to '2006-01-01 00:00:00.000'

lunarweekstart ('2006-01-12', 0, 1) returns '2006-01-09' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

LunarweekEnd(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a value corresponding to a timestamp with the last millisecond of the lunar week (consecutive 7 day periods starting on January 1st each year) containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the lunar week which contains *date*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

lunarweekend ('2006-01-12') returns '2006-01-14' with an underlying numeric value corresponding to '2006-01-14 23:59:59.999'

lunarweekend ('2006-01-12', -1) returns '2006-01-07' with an underlying numeric value corresponding to '2006-01-07 23:59:59.999'

lunarweekend ('2006-01-12', 0, 1) returns '2006-01-15' with an underlying numeric value corresponding to '2006-01-15 23:59:59.999'

LunarWeekName(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a display value showing the year and week number with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the lunar week (consecutive 7 day periods starting on January 1st each year) containing *date*. *Shift* is an integer, where the value 0 indicates the lunar week which contains *date*. Negative values in period indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

lunarweekname ('2006-01-12') returns '2006/02' with an underlying numeric value corresponding to '2006-01-08 00:00:00.000'

lunarweekname ('2006-01-12', -1) returns '2006/01' with an underlying numeric value corresponding to '2006-01-01 00:00:00.000'

lunarweekname ('2006-01-12', 0, 1) returns '2006/02' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

DayStart(*timestamp* [, *shift* = 0 [, *dayoffset* = 0]])

Returns a value corresponding to a timestamp with the first millisecond of the day contained in *timestamp*. The default output format will be the `TimestampFormat` set in the script. *Shift* is an integer, where the value 0 indicates the day which contains *date*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting mid-night, you may indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

Examples:

daystart ('2006-01-25 16:45') returns '2006-01-25 00:00:00' with an underlying numeric value corresponding to '2006-01-25 00:00:00.000'

daystart ('2006-01-25 16:45', -1) returns '2006-01-24 00:00:00' with an underlying numeric value corresponding to '2006-01-24 00:00:00.000'

daystart ('2006-01-25 16:45', 0, 0.5) returns '2006-01-25 12:00:00' with an underlying numeric value corresponding to '2006-01-25 12:00:00.000'

DayEnd(*timestamp* [, *shift* = 0 [, *dayoffset* = 0]])

Returns a value corresponding to a timestamp with the last millisecond of the day contained in *timestamp*. The default output format will be the `TimestampFormat` set in the script. *Shift* is an integer, where the value 0 indicates the day which contains *date*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting mid-night, you may indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

Examples:

dayend ('2006-01-25 16:45') returns '2006-01-25 23:59:59' with an underlying numeric value corresponding to '2006-01-25 23:59:59.999'

dayend ('2006-01-25 16:45', -1) returns '2006-01-24 23:59:59' with an underlying numeric value corresponding to '2006-01-24 23:59:59.999'

dayend ('2006-01-25 16:45', 0, 0.5) returns '2006-01-26 11:59:59' with an underlying numeric value corresponding to '2006-01-26 11:59:59.999'

DayName(*timestamp* [, *shift* = 0 [, *dayoffset* = 0]])

Returns a display value showing the date with an underlying numeric value corresponding to a timestamp with the first millisecond of the day containing *timestamp*. *Shift* is an integer, where the value 0 indicates the day which contains date. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

Examples:

dayname ('2006-01-25 16:45') returns '2006-01-25' with an underlying numeric value corresponding to '2006-01-25 00:00:00.000'

dayname ('2006-01-25 16:45', -1) returns '2006-01-24' with an underlying numeric value corresponding to '2006-01-24 00:00:00.000'

dayname ('2006-01-25 16:45', 0, 0.5) returns '2006-01-25' with an underlying numeric value corresponding to '2006-01-25 12:00:00.000'

setdateyear (*timestamp*, *year*)

returns a timestamp based on *timestamp* but with the year replaced with *year*.

timestamp is a standard QlikView timestamp (often just a date).

year is a four-digit year.

Examples:

setdateyear('2005-10-29', 2006) returns '2006-10-29'

setdateyear('2005-10-29 04:26', 2006) returns '2006-10-29 04:26'

setdateyearmonth (*timestamp*, *year*, *month*)

returns a timestamp based on *timestamp* but with the year replaced with *year* and the month replaced with *month*.

timestamp is a one- or two-digit month.

year is a four-digit year.

month is a one- or two-digit month.

Examples:

setdateyearmonth('2005-10-29', 2006, 3) returns '2006-03-29'

setdateyearmonth('2005-10-29 04:26', 2006, 3) returns '2006-03-29 04:26'

InYear (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the year containing *basedate*. The year can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the year which contains *basedate*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inyear ('2006-01-25', '2006-01-01', 0) returns true
inyear ('2005-01-25', '2006-01-01', 0) returns false
inyear ('2006-01-25', '2006-01-01', -1) returns false
inyear ('2005-01-25', '2006-01-01', -1) returns true
inyear ('2006-01-25', '2006-07-01', 0, 3) returns false
inyear ('2006-03-25', '2006-07-01', 0, 3) returns true

InYearToDate (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the part of year containing *basedate* up until and including the last millisecond of *basedate*. The year can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the year which contains *basedate*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inyeartodate ('2006-01-25', '2006-02-01', 0) returns true
inyeartodate ('2006-01-25', '2006-01-01', 0) returns false
inyeartodate ('2005-01-25', '2006-02-01', -1) returns true

InQuarter (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the quarter containing *basedate*. The quarter can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the quarter which contains *basedate*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inquarter ('2006-01-25', '2006-01-01', 0) returns true
inquarter ('2006-01-25', '2006-04-01', 0) returns false
inquarter ('2006-01-25', '2006-01-01', -1) returns false
inquarter ('2005-12-25', '2006-01-01', -1) returns true

inquarter ('2006-01-25', '2006-03-01', 0, 3) returns false
inquarter ('2006-03-25', '2006-03-01', 0, 3) returns true

InQuarterToDate (*date*, *basedate* , *shift* [, *first_month_of_year* = 1])
returns true if *date* lies inside the part of the quarter containing *basedate* up until and including the last millisecond of *basedate*. The quarter can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the quarter which contains *basedate*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inquartertodate ('2006-01-25', '2006-01-25', 0) returns true
inquartertodate ('2006-01-25', '2006-01-24', 0) returns false
inquartertodate ('2005-10-25', '2006-02-01', -1) returns true

InMonth (*date*, *basedate* , *shift*)

returns true if *date* lies inside the month containing *basedate*. The month can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the month which contains *basedate*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

inmonth ('2006-01-25', '2006-01-01', 0) returns true
inmonth ('2006-01-25', '2006-04-01', 0) returns false
inmonth ('2006-01-25', '2006-01-01', -1) returns false
inmonth ('2005-12-25', '2006-01-01', -1) returns true

InMonthToDate (*date*, *basedate* , *shift*)

returns true if *date* lies inside the part of month containing *basedate* up until and including the last millisecond of *basedate*. The month can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the month which contains *basedate*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

inmonthtodate ('2006-01-25', '2006-01-25', 0) returns true
inmonthtodate ('2006-01-25', '2006-01-24', 0) returns false
inmonthtodate ('2006-01-25', '2006-02-28', -1) returns true

InMonths (*n*, *date*, *basedate* , *shift* [, *first_month_of_year* = 1])
 returns true if *date* lies inside the *n* month period (aligned from January 1st) containing *basedate*. *N* must be (1), 2, (3), 4 or 6. The period can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the period which contains *basedate*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inmonths (4, '2006-01-25', '2006-01-01', 0) returns true
inmonths (4, '2006-01-25', '2006-05-01', 0) returns false
inmonths (4, '2006-01-25', '2006-01-01', -1) returns false
inmonths (4, '2005-12-25', '2006-01-01', -1) returns true
inmonths (4, '2006-01-25', '2006-03-01', 0, 3) returns false
inmonths (4, '2006-04-25', '2006-03-01', 0, 3) returns true

InMonthsToDate (*n*, *date*, *basedate* , *shift* [, *first_month_of_year* = 1])
 returns true if *date* lies inside the part of the *n* month period (aligned from January 1st) containing *basedate* up until and including the last millisecond of *basedate*. *N* must be (1), 2, (3), 4 or 6. The period can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the period which contains *basedate*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inmonthstodate (4, '2006-01-25', '2006-04-25', 0) returns true
inmonthstodate (4, '2006-04-25', '2006-04-24', 0) returns false
inmonthstodate (4, '2005-11-25', '2006-02-01', -1) returns true

InWeek (*date*, *basedate* , *shift* [, *weekstart*])
 returns true if *date* lies inside the week containing *basedate*. The week can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the week which contains *basedate*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekstart*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

inweek ('2006-01-12', '2006-01-14', 0) returns true
inweek ('2006-01-12', '2006-01-20', 0) returns false
inweek ('2006-01-12', '2006-01-14', -1) returns false
inweek ('2006-01-07', '2006-01-14', -1) returns true
inweek ('2006-01-12', '2006-01-09', 0, 3) returns false

InWeekToDate (*date*, *basedate*, *shift* [, *weekstart*])

returns true if *date* lies inside the part of week containing *basedate* up until and including the last millisecond of *basedate*. The week can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the week which contains *basedate*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekstart*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

inweektodate ('2006-01-12', '2006-01-12', 0) returns true
inweektodate ('2006-01-12', '2006-01-11', 0) returns false
inweektodate ('2006-01-12', '2006-01-05', -1) returns false

InLunarWeek (*date*, *basedate*, *shift* [, *weekstart*])

returns true if *date* lies inside the lunar week (consecutive 7 day periods starting on January 1st each year) containing *basedate*. The lunar week can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the lunar week which contains *basedate*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekstart*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

inlunarweek ('2006-01-12', '2006-01-14', 0) returns true
inlunarweek ('2006-01-12', '2006-01-20', 0) returns false
inlunarweek ('2006-01-12', '2006-01-14', -1) returns false
inlunarweek ('2006-01-07', '2006-01-14', -1) returns true
inlunarweek ('2006-01-11', '2006-01-08', 0, 3) returns false

InLunarWeekToDate (*date*, *basedate*, *shift* [, *weekstart*])

returns true if *date* lies inside the part of lunar week (consecutive 7 day periods starting on January 1st each year) containing *basedate*

up until and including the last millisecond of *basedate*. The lunar week can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the lunar week which contains *basedate*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekstart*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

inlunarweektodate ('2006-01-12', '2006-01-12', 0) returns true

inlunarweektodate ('2006-01-12', '2006-01-11', 0) returns false

inlunarweektodate ('2006-01-12', '2006-01-05', 1) returns true

InDay (*timestamp*, *basetimestamp*, *shift* [, *daystart*])

returns true if *timestamp* lies inside the day containing *basetimestamp*. The day can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the day which contains *basetimestamp*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *daystart*, e.g 0.125 to denote 3am.

Examples:

inday ('2006-01-12 12:23', '2006-01-12 00:00', 0) returns true

inday ('2006-01-12 12:23', '2006-01-13 00:00', 0) returns false

inday ('2006-01-12 12:23', '2006-01-12 00:00', -1) returns false

inday ('2006-01-11 12:23', '2006-01-12 00:00', -1) returns true

inday ('2006-01-12 12:23', '2006-01-12 00:00', 0, 0.5) returns false

inday ('2006-01-12 11:23', '2006-01-12 00:00', 0, 0.5) returns true

InDayToTime (*timestamp*, *basetimestamp*, *shift* [, *daystart*])

returns true if *timestamp* lies inside the part of day containing *basetimestamp* up until and including the exact millisecond of *basetimestamp*. The day can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the day which contains *basetimestamp*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *daystart*, e.g 0.125 to denote 3am.

Examples:

indaytotime ('2006-01-12 12:23', '2006-01-12 23:59', 0) returns true

indaytotime ('2006-01-12 12:23', '2006-01-12 00:00', 0) returns false

indaytotime ('2006-01-11 12:23', '2006-01-12 23:59', -1) returns true

timezone ()

returns the name of the current time zone as defined in Windows.

Example:

timezone ()

GMT () returns the current Greenwich Mean Time as derived from the system clock and Windows time settings.

Example:

gmt ()

UTC() returns the current Coordinated Universal Time.

Example:

utc()

daylightsaving ()

returns the current adjustment for daylight saving time as defined in Windows.

Example:

daylightsaving ()

age(*timestamp*, *date_of_birth*)

Returns the age at the time of *timestamp* (in completed years) of someone born on *date_of_birth*.

Example:

age('2007-01-25', '2005-10-29') returns 1

age('2007-10-29', '2005-10-29') returns 2

networkdays(*start_date*, *end_date* [, *holiday*])

Returns the number of working days (Monday-Friday) between and including *start_date* and *end_date* taking into account any optionally listed *holidays*. All parameters should be valid dates or timestamps.

Example:

networkdays ('2007-02-19', '2007-03-01') returns 9

networkdays ('2006-12-18', '2006-12-31', '2006-12-25', '2006-12-26') returns 8

firstworkdate(*end_date*, *no_of_workdays* {, *holiday*})

Returns the latest starting date to achieve *number_of_workdays* (Monday-Friday) ending no later than *end_date* taking into account any optionally listed holidays. *End_date* and *number_of_workdays* should be valid dates or timestamps.

Example:

firstworkdate ('2007-03-01', 9) returns '2007-02-19'

firstworkdate ('2006-12-31', 8, '2006-12-25', '2006-12-26') returns '2006-12-18'

lastworkdate(*start_date*, *no_of_workdays* {, *holiday*})

Returns the earliest ending date to achieve *number_of_workdays* (Monday-Friday) if starting at *start_date* taking into account any optionally listed holidays. *Start_date* and *number_of_workdays* should be valid dates or timestamps.

Example:

lastworkdate ('2007-02-19', 9) returns '2007-03-01'

lastworkdate ('2006-12-18', 8, '2006-12-25', '2006-12-26') returns '2006-12-29'

Number Interpretation Functions

The number interpretation functions are a set of functions to interpret a field content or an expression. With these functions it is possible to state the data type, decimal separator, thousand separator etc. used.

If no interpretation functions are used, QlikView interprets the data as a mix of numbers, dates, times, time stamps and strings, using the default settings for number format, date format and time format defined by script variables and by the operating system.

In order to understand the examples in this section, you should first study the chapter “Number Interpretation” on page 289.

Note For reasons of clarity, all number representations are given with decimal point as decimal separator.

num#(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **num#** function evaluates the *expression* numerically according to the string given as *format-code*. For a description of the format-code, see page 305. Decimal separator and thousand separator can be set as third and fourth parameters. If the parameters 2-4 are omit-

ted, the default number format set by script variables or in the operating system is used.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
number format	# ##0,#	#,##0.#
num# (A, '#') where A=35,648.375 returns		

Setting 1		Setting 2	
string	number	string	number
35,648.375	-	35648.375	35648.375

num#(A, '#.#', ',', '.') where A=35,648.375 returns

Setting 1 and 2	
string	number
35,648.375	-35648.375

num#(A, '#.#', ',', '.') where A=35648.375 returns

Setting 1 and 2		
string	number	
35648.375	35648375	Note! The number represent.!

num#(A, 'abc#,#') where A=abc123,4 returns

Setting 1		Setting 2	
string	number	string	number
abc123,4	123.4	abc123,4	1234

money#(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **money#** function evaluates the *expression* numerically according to the string given as *format-code*. For a description of the *format-code*, see page 305. Decimal separator and thousand separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the default number format set by script variables or in the operating system is used.

The **money#** function generally behaves just like the **num#** function but takes its default values for decimal and thousand separator from the script variables for money format or the system settings for currency.

Examples:

The examples below assume the two following default settings:

Default setting 1	Default setting 2
-------------------	-------------------

money format kr # ##0,00 \$ #,##0.00
money#(A, '# ##0,00 kr') where A=35 648,37 kr returns

Setting 1		Setting 2	
string	number	string	number
35 648.37 kr	35648.37	35 648.37 kr	3564837

money#(A, '\$#', '.', ',') where A=\$35,648.37 returns

Setting 1 and 2	
string	number
\$35,648.37	35648.37

date#(*expression* [, *format-code*])

The **date#** function evaluates the *expression* as a date according to the string given as *format-code*. For a description of the format-code, see page 305. If the format code is omitted, the default date format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
date# (A) where A=8/6/97 returns		

Setting 1		Setting 2	
string	number	string	number
8/6/97	-	8/6/97	35648

date#(A, 'YYYY.MM.DD') where A=1997.08.06 returns

Setting 1 and 2	
string	number
1997.08.06	35648

time#(*expression* [, *format-code*])

The **time#** function evaluates the *expression* as a time according to the string given as *format-code*. For a description of the format-code, see page 305. If the format code is omitted, the default time format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

Default setting 1	Default setting 2
-------------------	-------------------

time format hh:mm:ss hh.mm.ss
time#(A) where A=09:00:00 returns

Setting 1		Setting 2	
string	number	string	number
09:00:00	0.375	09:00:00	-

time#(A, 'hh.mm') where A=09.00 returns

Setting 1 and 2	
string	number
09.00	0.375

timestamp#(expression [, format-code])

The **timestamp#** function evaluates the *expression* as a date and a time according to the string given as *format-code*. For a description of the format-code, see page 305. If the format code is omitted, the default date and time formats set in the operating system are used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
time format	hh:mm:ss	hh:mm:ss

timestamp#(A) where A=8/6/97 09:00:00 returns

Setting 1		Setting 2	
string	number	string	number
8/6/97 09:00:00	-	8/6/97 09:00:00	35648.375

timestamp#(A, 'YYYY-MM-DD hh_mm') where A=1997 08-06 09_00 returns

Setting 1 and 2	
string	number
1997-08-06 09_00	35648.375

interval#(expression [, format-code])

The **interval#** function evaluates the *expression* as a time interval according to the string given as a *format-code*. For a description of the format-code, see page 305. If the format code is omitted, the time format set in the operating system is used.

The **interval#** function generally behaves just like the **time#** function but while times can never be greater than 23:59:59 (numeric

num(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **num** function formats the *expression* numerically according to the string given as *format-code*. For a description of the *format-code*, see page 305 in book II. Decimal separator and thousand separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the number format set in the operating system is used.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
number format	# ##0,#	#,##0.#
num (A , '0.0') where A=35648.375 returns		

Setting 1		Setting 2	
string	number	string	number
35 648 375	35648375	35648.375	35648.375

num(A , '#,##0.##' , ',' , '.') where A=35648 returns

Setting 1 and 2	
string	number
35,648.00	35648

num(*pi*() , '0,00') returns

Setting 1		Setting 2	
string	number	string	number
3,14	3.141592653	003	3.141592653

money(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **money** function formats the *expression* numerically according to the string given as *format-code*. For a description of the *format-code*, see page 305 in book II. Decimal separator and thousand separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the number format set in the operating system is used.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
money format	kr # ##0,00	\$ #,##0.00
money (A) where A=35648 returns		

Setting 1		Setting 2	
string	number	string	number
kr 35 648,00	35648.00	\$ 35,648.00	35648.00

money(A, '#,##0 -', ',', ',') where A=3564800 returns

Setting 1 and 2	
string	number
3,564,800 -	3564800

date(*expression* [, *format-code*])

The **date** function formats the *expression* as a date according to the string given as *format-code*. For a description of the format-code, see page 305 in book II. If the format code is omitted, the date format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
date (A) where A=35648 returns		

Setting 1		Setting 2	
string	number	string	number
97-08-06	35648	8/6/97	35648

date(A, 'YY-MM-DD') where A=35648 returns

Setting 1 and 2	
string	number
97-08-06	35648

date(A, 'DD.MM.YYYY') where A=35648.375 returns

Setting 1 and 2	
string	number
06.08.1997	35648.375

date(A, 'YY.MM.DD') where A=8/6/97 returns

Setting 1		Setting 2	
string	number	string	number
<NULL>	(nothing)	97.08.06	35648

time(*expression* [, *format-code*])

The **time** function formats the *expression* as a time according to the string given as *format-code*. For a description of the format-code, see page 305 in book II. If the format code is omitted, the time format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
time format	hh:mm:ss	hh.mm.ss

time(A) where A=0.375 returns

Setting 1		Setting 2	
string	number	string	number
09:00:00	0.375	09.00.00	0.375

time(A) where A=35648.375 returns

Setting 1		Setting 2	
string	number	string	number
09:00:00	35648.375	09.00.00	35648.375

time(A, 'hh-mm') where A=0.99999 returns

Setting 1 and 2	
string	number
23-59	0.99999

timestamp(expression [, format-code])

The **timestamp** function formats the *expression* as a date and a time according to the string given as *format-code*. For a description of the format-code, see page 305 in book II. If the format code is omitted, the date and time formats set in the operating system are used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
time format	hh:mm:ss	hh:mm:ss

timestamp(A) where A=35648.375 returns

Setting 1		Setting 2	
string	number	string	number
97-08-06	35648.375	8/6/97 09:00:00	35648.375
09:00:00			

timestamp(A, 'YYYY-MM-DD hh.mm') where A=35648 returns

Setting 1 and 2	
string	number

1997-08-06 00.00	35648
------------------	-------

interval(*expression* [, *format-code*])

The **interval** function formats the *expression* as a time interval according to the string given as a *format-code*. For a description of the format-code, see page 305 in book II. If the format code is omitted, the time format set in the operating system is used.

Intervals may be formatted as a time, as days or as a combination of days, hours, minutes, seconds and fractions of seconds.

Examples:

The examples below assume the following operating system settings:

short date format	YY-MM-DD
time format	hh:mm:ss
number decimal format	.

interval(A) where A=0.375 returns

string	number
09:00:00	0.375

interval(A) where A=1.375 returns

string	number
33:00:00	1.375

interval(A, 'D hh:mm') where A=1.375 returns

string	number
1 09:00	1.375

interval(A-B, 'D hh:mm') where A=97-08-06 09:00:00 and B=96-08-06 00:00:00 returns

string	number
365 09:00	365.375

dual(*s* , *x*)

Forced association of an arbitrary string representations with a given number representation *x*. In QlikView, when several pieces of data read into one field have different string representations but the same valid number representation, they will all share the first string representation encountered. The **dual** function is typically used early in the script, before other data is read into the field concerned,

in order to create that first string representation, which will be shown in list boxes etc.

Example:

```
load dual(string,numrep) as DayOfWeek
inline [string,numrep
Monday,0
Tuesday,1
Wednesday,2
Thursday,3
Friday,4
Saturday,5
Sunday,6];
load Date, weekday(Date) as DayOfWeek from
afile.csv;
```

The script example will generate a field *DayOfWeek* with the weekdays written in clear text. QlikView will for all purposes regard the field as a numeric field.

Color Functions

colormix1(Value, ColorZero, ColorOne)

returns an RGB color representation from a gradient between two colors based on a value between 0 and 1. If the value is equal to zero the first color will be returned. If it is 1, the second color is returned and if it is in between the appropriate intermediate shading is returned.

Value is a real number between 0 and 1.

ColorZero is a valid RGB color representation for the color to be associated with the low end of the interval.

ColorOne is a valid RGB color representation for the color to be associated with the high end of the interval.

Example:

```
colormix1( x, black (), red() )
```

colormix2(Value, ColorMinusOne, ColorOne [,ColorZero])

returns an RGB color representation from a gradient between two colors based on a value between -1 and 1 with the possibility to specify an intermediate color for the center position. If the value is equal to -1 the first color will be returned. If it is 1, the second color is returned and if it is in between the appropriate intermediate shading is returned.

Value is a real number between -1 and 1.

ColorMinusOne is a valid RGB color representation for the color to be associated with the low end of the interval.

ColorOne is a valid RGB color representation for the color to be associated with the high end of the interval.

ColorZero is an optional valid RGB color representation for the color to be associated with the center of the interval.

Examples:

colormix2(x, red (), green()) returns colors from red to green via brown

colormix2(x, red (), green(), black()) returns colors from red to green via black



23 DATA STRUCTURES

23.1 Data Loading Statements

Data is loaded by **load** or **select** statements. Each of these statements generates an input table. A table can always be seen as a list of something, each record (row) then being a new instance of the object type and each field (column) being a specific attribute or property of the object.

Rules:

QlikView does not make any difference between tables generated by a **load** or a **select** statement. Hence, if several tables are loaded, it does not matter whether the tables are loaded by **load** or **select** statements or by a mix of the two.

The order of the fields in the statement or in the original table in the database is arbitrary to the QlikView logic.

Field names are used in the further process to identify fields and making associations. These are case sensitive. It is thus often necessary to rename fields in the script. See “Renaming Fields” on page 466 .

23.2 Execution of the Script

For a typical **load** or **select** statement the order of events is roughly as follows:

- 1 Evaluation of expressions
- 2 Renaming of fields by **as**
- 3 Renaming of fields by **alias**
- 4 Qualification of table name, if applicable
- 5 Mapping of data if field name matches

23.3 System Fields

When extracting fields from a data source, QlikView also produces a number of system fields:

<i>\$Table</i>	Names of all logical tables loaded by the script.
<i>\$Field</i>	Names of the fields that are read from the tables.
<i>\$Fields</i>	Number of fields in different tables.

<i>\$FieldNo</i>	Position of the fields in the tables.
<i>\$Rows</i>	Number of rows in the tables.
<i>\$Info</i>	Names of info tables included in the document.

The system fields can be displayed as list boxes in a QlikView document. With the help of these list boxes, you can easily understand the underlying structure of the application.

When a single table is selected in the list box *\$Table*, an information symbol is shown in the title bar of the list box. By clicking here, you may view the table (if it is a text file).

By setting the *\$Field* list box to **Show Frequency** in the **General** page of the **List Box Properties** dialog, it is easy to see which fields are associating the different tables.

Another very useful tool is a pivot table with the two dimensions *\$Field* and *\$Table* and the expression only(*\$Field*). Put the *\$Field* dimension to the left and the *\$Table* dimension on the top. Sort both dimensions by load order. You will now have an excellent system table showing the relations of tables and fields.

You create a System table by choosing **New Sheet Object, System Table** from the **Layout** menu.

23.4 Logical Tables

Each **load** or **select** statement generates a table. Normally, QlikView treats the result of each one of these as *one logical table*. However, there are a couple of exceptions from this rule:

- If two or more statements result in tables with identical field names, the tables are concatenated and treated as *one logical table*.
- If a **load** or **select** statement is preceded by any of the following qualifiers, data is altered or treated differently:

concatenate

this table is added (concatenated) to the last previously created logical table.

crosstable

this table is converted from crosstable format to column format.

generic

this table is split into several other logical tables.

info

this table is loaded not as a logical table, but as an information table containing links to external info such as files, sounds, URLs, etc.

intervalmatch

the table (which must contain exactly two columns) is interpreted as numeric intervals, which are associated with discrete numbers in a specified field.

join

this table is joined by QlikView with the previously created logical table, over the fields in common.

mapping

this table (which must contain exactly two columns) is read as a mapping table, which is never associated with other tables.

semantic

this table is loaded not as a logical table, but as a *semantic table* containing relationships that should not be joined, e.g. predecessor, successor and other references to other objects of the same type.

When the data has been loaded, the logical tables are associated. The logical tables and the associations can be seen in the Table Viewer dialog, see page 228.

23.5 Associations between Logical Tables

Data Associations

A database can have many tables. Each table can be considered as a list of something, i.e. each record in the list represents an instance of an object of some type.

If two tables are lists of different things, e.g. if one is a list of customers and the other a list of invoices, and the two tables have a field in common, e.g. the customer number, this is usually a sign that there is a relationship between the two tables. In standard SQL query tools the two tables should almost always be *joined*.

The tables defined in the QlikView script are called logical tables. QlikView makes *associations* between the tables based on the field names, and performs the joins when a selection is made, i.e. when the user clicks on a list box item.

Hence, a QlikView association is almost the same thing as a QlikView join (see below). The only difference is that the join is performed when the script is executed - the logical table is usually the result of the join. The association is made after the logical table is created - associations are always made between the logical tables.

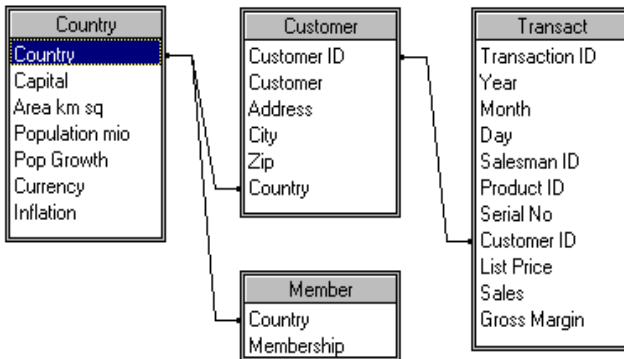


Figure 93. Four tables: a list of countries, a list of customers, a list of transactions and a list of memberships, which are associated with each other through the fields Country and CustomerID.

A QlikView association has approximately the same effect as a SQL natural outer join. The QlikView association is however more general: an outer join in SQL is usually a one way projection of one table on another. A QlikView association always results in a full (bidirectional) natural outer join.

Frequency Information in Associating Fields

There are some limitations in the use of most associating fields, i.e. fields which are common between two or more tables. When a field occurs in more than one table, QlikView has a problem knowing which of the tables it should use for calculating data frequencies.

QlikView analyzes the data to see if there is a non-ambiguous way to identify a main table to count in (sometimes there is) but in most cases the program can only make a guess. Since an incorrect guess could be fatal (QlikView would appear to make a calculation error) the program has been designed not to allow certain operations when the data interpretation is ambiguous for associating fields.

For associating fields the following limitations normally apply:

- 1 You cannot display frequency information in a list box showing the field. The **Show Frequency** option in the **List Box Properties (General)** page is dimmed.
- 2 Statistics boxes for the field show n/a for most statistical entities.
- 3 In charts it is not possible to create expressions containing functions depending on frequency information (sum, count functions, average etc.) on the field, unless the **Distinct** modifier is turned on. If you try, an error message will appear. After each reload, QlikView will scan all chart expressions to see if any ambiguities have occurred as a result of changes in data structures. If ambiguous expressions are found, a warning dialog will be shown and the expression will be disabled. It will not be possible to enable the expression until the problem has been corrected. If a log file is enabled, all ambiguous expressions will be listed in the log.

There is a simple way to overcome these limitations. Load the field an extra time under a new name from the table where frequency counts should be made. Then use the new field for a list box with frequency, for a statistics box or for calculations in your charts.

Synthetic Keys

When two or more input tables have two or more fields in common, this implies a composite key relationship. QlikView handles this through synthetic keys. These keys are anonymous fields that represent all occurring combinations of the composite key. When the number of composite keys increases, depending on data amounts, table structure and other factors, QlikView may or may not handle them gracefully. QlikView may end up using excessive amount of time and/or memory. Unfortunately the actual limitations are virtually impossible to predict, which leaves only trial and error as a practical method to determine them.

Therefore we recommend an overall analysis of the intended table structure by the application designer. Typical tricks include:

- Forming your own non-composite keys, typically using string concatenation inside an **AutoNumber** script function.
- Making sure only the necessary fields connect. If you for example use a date as a key, make sure you do not load e.g. *year*, *month* or *day_of_month* from more than one input table.

Circular References ("Loops")

If there are circular references ("loops") in a data structure, the tables are associated in such a way that there is more than one path of associations between two fields.

This type of data structure should normally be avoided as much as possible, since it might lead to ambiguities in the interpretation of data. Unfortunately, circular structures are quite common in the real world. In some cases they are a result of poor database design, but in certain cases they may be unavoidable.

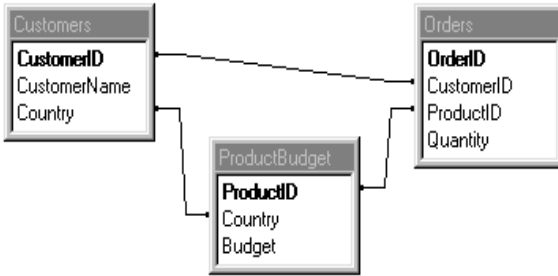


Figure 94. Three tables with a circular reference.

QlikView solves the problem of circular references by breaking the loop with a loosely coupled table. When QlikView finds circular data structures while executing the load script, a warning dialog will be shown and one or more tables will be set as loosely coupled. QlikView will typically attempt to loosen the longest table in the loop, as this is often a transaction table, which normally should be the one to loosen. If you do not like QlikView's default choice of table to set as loosely coupled, it is possible to declare a specific table as loosely coupled via a **loosen table** statement in the script. It is also possible to change the settings for loosely coupled tables interactively in the **Tables** page of **Document Properties** after the script has been executed. See chapter "Loosely Coupled Tables" on page 144, for more information on loosely coupled tables.

23.6 Renaming Fields

Sometimes it is necessary to rename fields in order to obtain the correct associations.

It could be that two fields are named differently although they denote the same thing, e.g. *ID* in a table *Customers* and *CustomerID* in a table *Orders*. Obviously they both denote a specific customer identification code and should both be named *CustomerID* or something similar.

It could also be that two fields have the same name but denote different things, e.g. *Date* in the table *Invoices* and *Date* in the table *Orders*. These should preferably be renamed *InvoiceDate* and *OrderDate* or something similar.

There could also be plain misspellings in the database or different conventions on upper and lower case letters. (Since QlikView differs between upper and lower case letters it is important that such things are corrected.)

Fields can be renamed in the script, so there is no need to change the original data. There are three ways of renaming fields:

- The **load** or **select** statement can be preceded by an **alias** statement.

Example:

```
Alias ID as CustomerID;
Load * from Customer.csv;
```

- The **load** or **select** statement can contain the **as** specifier (automatically generated if the file wizard is used).

Example:

```
Load ID as CustomerID, Name, Address, Zip, City,
State from Customer.csv;
```

- The **rename field** statement can be used to rename fields that are already loaded. Optionally, a mapping table containing one or more pairs of old and new field names can be used.

Example:

```
Load Number, Name, Street, Zip, City, Country
From Customers.csv;
rename field Number to Customer number;
```

Finally it is also possible to rename list boxes and other sheet objects, without changing the logical associations defined by the field names. Choose **Properties** from the object menu of the sheet object.

23.7 Concatenating Several Tables into One

Automatic Concatenation

If the field names and the number of fields of two or more loaded tables are exactly the same, QlikView will automatically concatenate the content of the different statements into one table.

Example:

```
load a, b, c from table1.csv;  
load a, c, b from table2.csv;
```

The resulting logical table has the fields a, b and c. The number of records is the sum of the numbers of records in table 1 and table 2.

Rules:

- The number and names of the fields must be exactly the same.
- The order of the two statements is arbitrary.

Forced Concatenation

If two or more tables do not have exactly the same set of fields, it is still possible to force QlikView to concatenate the two tables. This is done with the **concatenate** prefix in the script, which concatenates a table with another named table or with the last previously created logical table.

Example:

```
load a, b, c from table1.csv;  
concatenate load a, c from table2.csv;
```

The resulting logical table has the fields a, b and c. The number of records in the resulting table is the sum of the numbers of records in table 1 and table 2. The value of field b in the records coming from table 2 is NULL.

Rules:

- The names of the fields must be exactly the same.
- Unless a table name of a previously loaded table is specified in the **concatenate** statement the **concatenate** prefix uses the last previously created logical table. The order of the two statements is thus *not* arbitrary.

Preventing Concatenation

If the field names and the number of fields of two or more loaded tables are exactly the same, QlikView will automatically concatenate the content of the different statements into one table. This is possible to prevent with a **non-concatenate** statement. The table loaded with the associated **load** or **select** statement will then not be concatenated with the existing table.

Example:

```
load a, b, c from table1.csv;  
nonconcatenate load a, b, c from table2.csv
```

23.8 Join and Keep

It is possible to join tables already in the script. The QlikView logic will then not see the separate tables, but rather the result of the join, which is a single logical table. Sometimes this is preferable, sometimes even necessary, but most of the time it is not advisable, since:

- The loaded tables often become larger, and QlikView works slower.
- Some information may be lost: the frequency (number of records) within the original table may no longer be available.

The **keep** functionality, which has the effect of reducing one or both of the two tables to the intersection of table data before the tables are stored in QlikView, has been designed to reduce the number of cases where you need to use explicit joins. For an illustration of the difference between **join** and **keep**, see the examples under “Join and Keep” on page 469.

Note In this manual the term join is usually used for joins made before the logical tables are created. The *association*, made after the logical tables are created, is however essentially also a join.

Joins within an SQL Select Statement

With some ODBC drivers it is possible to make a join within the **select** statement. This is almost equivalent to making a join using the **join** prefix.

However, most ODBC drivers are not able to make a full (bidirectional) outer join. They are only able to make a left or a right outer join. A left (right) outer join only includes combinations where the joining key exists in the left (right) table. A full outer join includes any combination. QlikView automatically makes a full outer join.

Further, making joins in **select** statements is far more complicated than making joins in QlikView.

Example:

```
SELECT DISTINCTROW
  [Order Details].ProductID, [Order Details].
  UnitPrice, Orders.OrderID, Orders.OrderDate,
  Orders.CustomerID
FROM Orders
  RIGHT JOIN [Order Details] ON Orders.OrderID =
  [Order Details].OrderID;
```

This **select** statement joins a table containing orders to a fictive company, with a table containing order details. It is a right outer join, meaning that all

the records of *OrderDetails* are included, also the ones with an *OrderID* that does not exist in the table *Orders*. Orders that exist in *Orders* but not in *OrderDetails* are however not included.

Join

The simplest way to make a join is with the **join** prefix in the script, which joins the input table with the last previously created logical table. The join will be an outer join, creating all possible combinations of values from the two tables.

Example:

```
load a, b, c from table1.csv;
join load a, d from table2.csv;
```

The resulting logical table has the fields a, b, c and d. The number of records differs depending on the field values of the two tables.

Rules:

- The names of the fields to join over must be exactly the same.
- The number of fields to join over is arbitrary. Usually the tables should have one or a few fields in common. No field in common will render the cartesian product of the tables. All fields in common is also possible, but usually makes no sense.
- Unless a table name of a previously loaded table is specified in the **join** statement the **join** prefix uses the last previously created logical table. The order of the two statements is thus *not* arbitrary.

Keep

The explicit **join** prefix in the QlikView script language performs a full join of the two tables. The result is one table. In many cases such joins will result in very large tables. One of the main features of QlikView is its ability to make associations between tables instead of joining them, which reduces space in memory, increases speed and gives enormous flexibility. The **keep** functionality has been designed to reduce the number of cases where you need to use explicit joins.

The **keep** prefix between two **load** or **select** statements has the effect of reducing one or both of the two tables to the intersection of table data before they are stored in QlikView. The **keep** prefix must always be preceded by one of the keywords **inner**, **left** or **right**. The selection of records from the tables is made in the same way as in a corresponding join. However, the two tables are not joined and will be stored in QlikView as two separately named tables.

Inner

The **join** and **keep** prefixes in the QlikView script language can be preceded by the prefix **inner**.

If used before **join**, it specifies that the join between the two tables should be an inner join. The resulting table contains only combinations between the two tables with a full data set from both sides.

If used before **keep**, it specifies that the two tables should be reduced to their common intersection before being stored in QlikView.

Example:

Table1

A	B
1	aa
2	cc
3	ee

Table2

A	C
1	xx
4	yy

```
QVTable:
Select * from Table1;
inner join select * from Table2;
```

QVTable

A	B	C
1	aa	xx

```
QVTab1:
Select * from Table1;
QVTab2:
inner keep select * from Table2;
```

QVTab1

A	B
1	aa

QVTab2

A	C
1	xx

The two tables in the keep example are of course associated via the field A.

Left

The **join** and **keep** prefixes in the QlikView script language can be preceded by the prefix **left**.

If used before **join**, it specifies that the join between the two tables should be a left join. The resulting table only contains combinations between the two tables with a full data set from the first table.

If used before **keep**, it specifies that the second table should be reduced to its common intersection with the first table before being stored in QlikView.

Example:

Table1

A	B
1	aa
2	cc
3	ee

Table2

A	C
1	xx
4	yy

```
QVTable:  
Select * from Table1;  
left join select * from Table2;
```

QVTable

A	B	C
1	aa	xx
2	cc	-
3	cc	-

```
QVTab1:  
Select * from Table1;  
QVTab2:  
left keep select * from Table2;
```

QVTab1

A	B
1	aa
2	cc
3	ee

QVTab2

A	C
1	xx

The two tables in the keep example are of course associated via the field A.

Right

The **join** and **keep** prefixes in the QlikView script language can be preceded by the prefix **right**.

If used before **join**, it specifies that the join between the two tables should be a right join. The resulting table only contains combinations between the two tables with a full data set from the second table.

If used before **keep**, it specifies that the first table should be reduced to its common intersection with the second table before being stored in QlikView.

Example:

Table1

A	B
1	aa
2	cc
3	ee

Table2

A	C
1	xx
4	yy

```
QVTable:
Select * from Table1;
right join select * from Table2;
```

QVTable

A	B	C
1	aa	xx
4	-	yy

```
QVTab1:
Select * from Table1;
QVTab2:
right keep select * from Table2;
```

QVTab1

A	B
1	aa

QVTab2

A	C
1	xx
4	yy

The two tables in the **keep** example are of course associated via the field A.



24 EVALUATING THE LOADED DATA

24.1 Generic Databases

A generic database is a table in which the field names are stored as field values in one column, while the field values are stored in a second. Generic databases are usually used for attributes of different objects.

Consider the example to the right. It is a generic database containing two objects, a ball and a box. Obviously some of the attributes, like color and weight, are common to both the objects, while others, like diameter, height length and width are not.

Generic database		
object	attribute	value
ball	color	red
ball	diameter	10 cm
ball	weight	100 g
box	color	black
box	height	16 cm
box	length	20 cm
box	weight	500 g
box	width	10 cm

This is a typical generic database. On one hand it would be awkward to store the data in a way giving each attribute a column of its own, since many of the attributes are not relevant for a specific object.

On the other hand, it would look messy displaying it in a way that mixed lengths, colors and weights.

QlikView solves this problem in an elegant way. If data is stored in the compact way shown above, you can choose between the two different ways of displaying the data. QlikView automatically creates several logical tables from the generic database.

object	attribute	value
ball	color	black
ball	diameter	red
ball	height	10 cm
ball	length	16 cm
ball	weight	20 cm
ball	width	100 g
ball		500 g

If this table is loaded the standard way, we can get three different list boxes on the screen. However, if the table is loaded as a generic database, QlikView will split up column two and three into different list boxes. QlikView will then generate one field for each unique value of the second column.

object	color	weight	diameter	length	width	height
ball	black	100g	10 cm	20 cm	10 cm	16 cm
box	red	500g				

The syntax for doing this is simple:

Example:

```
Generic select * from GenericTable;
```

It does not matter whether a **load** or **select** statement is used to load the generic database.

24.2 Cross Tables

A cross table is a common type of table featuring a matrix of values between two orthogonal lists of header data. It could look like the table below.

Example 1:

ex1.csv

Year	Jan	Feb	Mar	Apr	May	Jun
1991	45	65	78	12	78	22
1992	11	23	22	22	45	85
1993	65	56	22	79	12	56
1994	45	24	32	78	55	15
1995	45	56	35	78	68	82

If this table is simply loaded into QlikView, the result will be one field for *Year* and one field for each of the months. This is generally not what you would like to have. One would probably prefer to have three fields generated, one for each header category (*Year* and *Month*) and one for the data values inside the matrix.

This can be achieved by adding the **crossstable** prefix to your **load** or **select** statement.

The statement for loading this cross table could be:

```
crossstable (Month, Sales) load * from ex1.csv;
```

The result in QlikView would be as follows:

Year	Month	Sales		
1991	Apr	11	45	85
1992	Feb	12	55	85
1993	Jan	15	56	65
1994	Jun	22	65	68
1995	Mar	23	68	78
	May	24	78	79
		32	79	82
		35	82	82

The cross table is often preceded by a number of qualifying columns, which should be read in a straightforward way. This is the case in example 2:

Example 2:

ex2.csv

Salesman	Year	Jan	Feb	Mar	Apr	May	Jun
A	1991	45	65	78	12	78	22
A	1992	11	23	22	22	45	85
A	1993	65	56	22	79	12	56
A	1994	45	24	32	78	55	15
A	1995	45	56	35	78	68	82
B	1991	57	77	90	24	90	34
B	1992	23	35	34	34	57	97
B	1993	77	68	34	91	24	68
B	1994	57	36	44	90	67	27
B	1995	57	68	47	90	80	94

In this case there are two qualifying columns to the left, followed by the matrix columns. The number of qualifying columns can be stated as a third parameter to the **crosstable** prefix as follows:

```
crosstable (Month, Sales, 2) load * from ex2.csv ;
```

The result in QlikView would be:

The screenshot shows a QlikView pivot table with the following dimensions and data:

Salesman	Year	Month	Sales			
A	1991	Apr	11	34	57	82
A	1992	Feb	12	35	65	85
A	1993	Jan	15	36	67	90
A	1994	Jun	22	44	68	91
A	1995	Mar	23	45	77	94
B		May	24	47	78	97
			27	55	79	
			32	56	80	

For a description of the syntax see page 291.

24.3 Matching Intervals to Discrete Data

The **intervalmatch** prefix to a **load** or **select** statement is used to link discrete numeric values to one or more numeric intervals. This is a very powerful feature which can be used e.g. in production environments as shown in the example below.

Example:

Look at the two tables below. The first table shows the start and end of production of different orders. The second table shows some discrete events. How can we associate

the discrete events with the orders, so that we know e.g. which orders were affected by the disturbances and which orders were processed by which shifts?

OrderLog:

START	END	ORDER
01:00	03:35	A
02:30	07:58	B
03:04	10:27	C
07:23	11:43	D

EventLog:

TIME	EVENT	COMMENT
00:00	0	Start of shift 1
01:18	1	Line stop
02:23	2	Line restart 50%
04:15	3	Line speed 100%
08:00	4	Start of shift 2
11:43	5	End of production

First load the two tables as usual and then link the field *Time* to the intervals defined by the fields *Start* and *End*:

```
Select * from OrderLog;
Select * from EventLog;
Intervalmatch (Time) select Start,End from OrderLog;
```

A table box in QlikView could now be created as below:

Time	Event	Comment	Order	Start	End
00:00	0	Start of shift 1	-	-	-
01:18	1	Line stop	A	01:00	03:35
02:23	2	Line restart 50%	A	01:00	03:35
04:15	3	Line speed 100%	B	02:30	07:58
04:15	3	Line speed 100%	C	03:04	10:27
08:00	4	Start of shift 2	C	03:04	10:27
08:00	4	Start of shift 2	D	07:23	11:43
11:43	5	End of production	D	07:23	11:43

We can now easily see that mainly order *A* was affected by the line stop but that the reduced line speed affected also order *B* and *C*. Only the orders *C* and *D* were partly handled by shift 2.

Please note the following points when using **intervalmatch**:

- Before the **intervalmatch** statement, the field containing the discrete data points (*Time* in the example above) must already have been read into QlikView. The **intervalmatch** statement does not read this field from the database table!
- The table read in the **intervalmatch load** or **select** statement must always contain exactly two fields (*Start* and *End* in the example above). In order to establish a link to other fields you must read the interval fields together with additional fields in a separate **load** or **select** statement (the first **select** statement in the example above).
- The intervals are always closed, i.e. the end points are included in the interval. A lower or upper limit denoted by a NULL value is considered to be minus infinity and infinity, respectively. Other non-numeric limits render the interval to be disregarded.
- The intervals may be overlapping and the discrete values will be linked to all matching intervals.

24.4 Using the Extended IntervalMatch Syntax to Resolve Slowly Changing Dimension Problems.

The extended **intervalmatch** syntax can be used for handling of the well-known problem of slowly changing dimensions in source data.

Sample script:

```
SET NullInterpret='';
IntervalTable:
Load Key, ValidFrom, Team from IntervalTable.xls;
NullAsValue FirstDate,LastDate;
Key:
Load
    Key,
    ValidFrom as FirstDate,
    date(if(Key=previous(Key) ,
    previous(ValidFrom) - 1)) as LastDate,
    Team
resident IntervalTable order by Key, ValidFrom desc;
drop table IntervalTable;
Transact:
Load Key, Name, Date, Sales from Transact.xls;
inner join intervalmatch (Date,Key) load FirstDate,
LastDate, Key resident Key;
```

Comments to the example above:

The statement

```
SET NullInterpret='';
```

is only required when reading data from a table file since missing values are defined as empty strings instead of nullvalues.

Loading the data from IntervalTable would result in the following table:

Key	FirstDate	Team
000110		Northwest
000110	2001-01-21	Southwest
000120		Northwest
000120	2003-01-06	Southeast
000120	2003-03-05	Northwest

The **nullasvalue** statement allows null values to map to the listed fields.

Create *Key*, *FirstDate*, *LastDate*, (attribute fields) by using **previous** and **order by** and thereafter the IntervalTable is dropped having been replaced by this key table.

Loading the data from Transact would result in the following table:

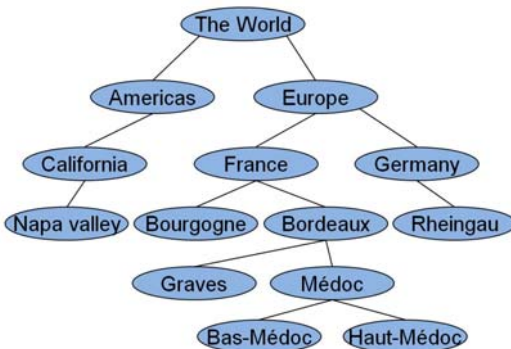
Key	Name	Date	Sales
000110	Spengler Aaron	1999-08-18	100
000110	Spengler Aaron	1999-12-25	200
000110	Spengler Aaron	2001-02-03	300
000110	Spengler Aaron	2001-05-05	400
000120	Ballard John	2001-06-04	500
000120	Ballard John	2003-01-20	600
000120	Ballard John	2003-03-10	700
000120	Ballard John	2003-03-13	800
000120	Ballard John	2003-09-21	900

The **intervalmatch** statement preceded by the **inner join** replaces the key above with a synthetic key that connects to the Transact table resulting in the following table:

Key	Team	Name	FirstDate	LastDate	TransactDate	Sales
000110	Northwest	Spengler Aaron		2001-01-20	1999-08-18	100
000110	Northwest	Spengler Aaron		2001-01-20	1999-12-25	200
000110	Southwest	Spengler Aaron	2001-01-21		2001-02-03	300
000110	Southwest	Spengler Aaron	2001-01-21		2001-05-05	400
000120	Northwest	Ballard John		2003-01-05	2001-06-04	500
000120	Northwest	Ballard John	2003-03-05		2003-03-10	700
000120	Northwest	Ballard John	2003-03-05		2003-03-13	800
000120	Northwest	Ballard John	2003-03-05		2003-09-21	900
000120	Southeast	Ballard John	2003-01-06	2003-03-04	2003-01-20	600

24.5 Hierarchies

Unbalanced n -level hierarchies are often used to represent e.g. geographical or organizational dimensions in data. These types of hierarchies are usually stored in an adjacent nodes table, i.e. in a table where each record corresponds to a node and has a field that contains a reference to the parent node.



NodeID	ParentID	NodeName
1	-	The World
2	1	Europe
3	2	France
4	3	Bordeaux
5	4	Medoc
6	5	Bas-Médoc
7	5	Haut-Médoc
8	4	Graves
9	3	Bourgogne
10	2	Germany
11	10	Rheingau
12	1	Americas
13	12	California
14	13	Napa valley

In such a table the node is stored on one record only but can still have any number of children. The table may of course contain additional fields describing attributes for the nodes.

An adjacent nodes table is optimal for maintenance, but difficult to use in everyday work. Instead, in queries and analysis, other representations are used. The expanded nodes table is one common representation, where each level in the hierarchy is stored in a separate field. The levels in an expanded nodes table can easily be used e.g. in a pivot table or in a tree structure. The **hierarchy** keyword can be used in the QlikView script to transform an adjacent nodes table to an expanded nodes table. See “Hierarchy” on page 299 for more information.

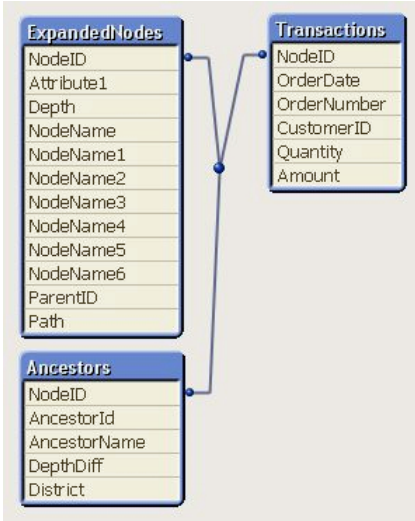
NodeID	ParentID	NodeName	Level0	Level1	Level2	Level3	Level4	Level5
1	-	The World	The World	-	-	-	-	-
2	1	Europe	The World	Europe	-	-	-	-
3	2	France	The World	Europe	France	-	-	-
4	3	Bordeaux	The World	Europe	France	Bordeaux	-	-
5	4	Medoc	The World	Europe	France	Bordeaux	Medoc	-
6	5	Bas-Médoc	The World	Europe	France	Bordeaux	Medoc	Bas-Médoc
7	5	Haut-Médoc	The World	Europe	France	Bordeaux	Medoc	Haut-Médoc
8	4	Graves	The World	Europe	France	Bordeaux	Graves	-
9	3	Bourgogne	The World	Europe	France	Bourgogne	-	-
10	2	Germany	The World	Europe	Germany	-	-	-
11	10	Rheingau	The World	Europe	Germany	Rheingau	-	-
12	1	Americas	The World	Americas	-	-	-	-
13	12	California	The World	Americas	California	-	-	-
14	13	Napa valley	The World	Americas	California	Napa valley	-	-

A problem with the expanded nodes table is that you cannot easily use the level fields for searches or selections, since you need à priori knowledge about which level you should search or select in. The ancestors table is a different representation that solves this problem. This representation is also called a bridge table.

NodeID	NodeName	AncestorID	AncestorName
1	The World	1	The World
2	Europe	1	The World
2	Europe	2	Europe
3	France	1	The World
3	France	2	Europe
3	France	3	France
4	Bordeaux	1	The World
4	Bordeaux	2	Europe
4	Bordeaux	3	France
4	Bordeaux	4	Bordeaux
5	Medoc	1	The World
5	Medoc	2	Europe
5	Medoc	3	France
5	Medoc	4	Bordeaux

The ancestors table contains one record for every child-ancestor relation found in the data. It contains keys and names for the children as well as for the ancestors. I.e. every record describes which node a specific node belongs to. The **hierarchybelongsto** keyword can be used in the QlikView script to transform an adjacent nodes table to an ancestors table. See “HierarchyBelongsTo” on page 300 for more information.

A good QlikView solution for a hierarchy needs both an expanded nodes table and an ancestors table. The former is needed to create pivot tables and generally describe the nodes; the latter to allow selection of entire trees. The two are linked through the node key, e.g. **NodeID**, which also links to a possible transaction table.



24.6 Semantic Links

Normally selections are made explicitly by clicking on the field values that are interesting. There is, however, also a way to make selections indirectly through semantic links. These are similar to field values, but with the difference that they describe the relations between the objects rather than the objects themselves. They appear as a list of buttons.

When you click on a semantic link, a selection is made in an other field.

Rules for Semantic Tables

Semantic links are created by loading tables containing the relations between the objects.

- The table must contain exactly three or four columns.

- A semantic table must either contain relations between field values of different fields or between field values of the same field. A mixture between the two is not accepted.
- The **load** or **select** statement loading a semantic table must be preceded by a **semantic** qualifier to show that it is not a logical table.

Normally four columns are used, the first one containing the field values that have a relation to some other field value and the third one containing the related field value. The second column must contain the names of the relations, and finally, the fourth one must contain the names of the inverse relations.

If three columns are used, no explicit names for the inverse relations can be given. The names given in the second column are used both for the relation and the inverse relation. The names are then preceded or followed by arrows.

Relation
<- Next
Next ->

If the relations are between field values of *the same* field, the first and third columns must have the same name. Also the names of the second and fourth column, i.e. the type of the relations, must be the same. However, if the relations are between field values of *different* fields, all columns must have different names.

Example: Extracting a semantic table from data

The semantic table does not always have to exist as a table outside QlikView. It is more flexible to extract this table from the existing table of objects through a separate **load** statement. In the *presidents* example in the QlikView examples directory, the script to generate the links *Predecessor* and *Successor* could be:

Relation
Predecessor
Successor

```
Directory presidents;
Load * from presdnts.csv (ansi, txt, delimiter
is ',', embedded labels);
Semantic Load
    No -1 as No,
    'Successor' as Relation,
    No,
    'Predecessor' as Relation
from presdnts.csv (ansi, txt, delimiter is ',',
embedded labels) where No > 1;
```

The second **load** statement results in a table that looks like the one to the right, and this table is loaded as a semantic table. The **where** clause is used to omit the first record since this would link the first president to the nonexistent 0:th president.

Semantic table			
No	Relation	No	Relation
1	Successor	2	Predecessor
2	Successor	3	Predecessor
3	Successor	4	Predecessor
4	Successor	5	Predecessor
5	Successor	6	Predecessor
6	Successor	7	Predecessor
7	Successor	8	Predecessor
8	Successor	9	Predecessor
9	Successor	10	Predecessor
10	Successor	11	Predecessor

Note also that this **load** statement contains two fields labeled *No* and two fields labeled *Relation*. Such a **load** statement would cause a script execution error if used to load a logical table since the load procedure for one single logical table demands that none of the fields have the same name. The corresponding **select** statement is also not possible, since most ODBC drivers also demand this. Instead, the following structure should be used if the presidents table is in a database:

```

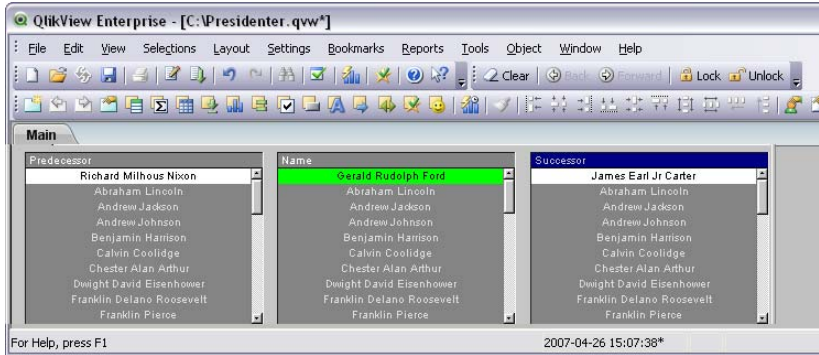
Connect to DataBase;
Select * from presdnts;
Alias No2 as No, Relation2 as Relation;
Semantic Select
    No -1 as No,
    'Successor' as Relation,
    No as No2,
    'Predecessor' as Relation2
from presdnts where No > 1;

```

The presidents example is just one simple example of how to use semantic links. These can also be used in genealogy, where the semantic links can be e.g. *cousin*, *sibling*, *grandmother*, etc. or for people in companies where the semantic links can be e.g. *superior*, *reports to*, *secretary*, etc.

Example: Using the related values as relation names

Sometimes it is more descriptive to use the related field value as name of the relation. In the case of the presidents, you may want all the predecessors in one column and all the successors in another:



To create these links, you would need the following script:

```
Load
    No as DuplicateOfNo,
    FirstName & ' ' & LastName as Name,
    *
from presdnts.csv;

Semantic Load
    No -1 as No,
    FirstName & ' ' & LastName as Successor,
    No as DuplicateOfNo,
    'Dummy1'
from presdnts.csv where No > 1;

Semantic Load
    No +1 as No,
    FirstName & ' ' & LastName as Predecessor,
    No as DuplicateOfNo,
    'Dummy2'
from presdnts.csv;
```

When a semantic link is clicked, a selection is made in the field of the third column, *DuplicateOfNo*, which in the semantic table is always the number of the president shown on the semantic link.

It may not be obvious at first, but the inverse relations in the above construction are almost useless. They would show a name of a president and, when

clicked, select the predecessor/successor of the shown president. This is why they are called *Dummy1* and *Dummy2* and only the first relation (column two) is used.

Since we do not want the dummy relations to appear in the list boxes, we must treat the second and fourth columns as different types of relations. This means that the first and third columns must have different field names. This is the reason why we have two columns containing the number of the president, *No* and *DuplicateOfNo*.

Two different **semantic** statements are needed since we want two different list boxes with relations.

This example can also be made with three-column semantic tables, but then the list boxes with the inverse relations will most likely confuse the user.

24.7 Linking Information to Field Values

Information in the form of text files, images or external application files can be associated to data in a QlikView document. To use this feature you will have to create tables describing what information file is to be linked to which field value, and tell QlikView to treat these tables as information tables. How this is done is explained below.

Information tables must consist of two columns, the first one headed by a field name and containing a list of values belonging to the field, the second one headed by an arbitrary name and containing the information (if text) or references to the files containing the information (images, applications). See Figure 95 below.

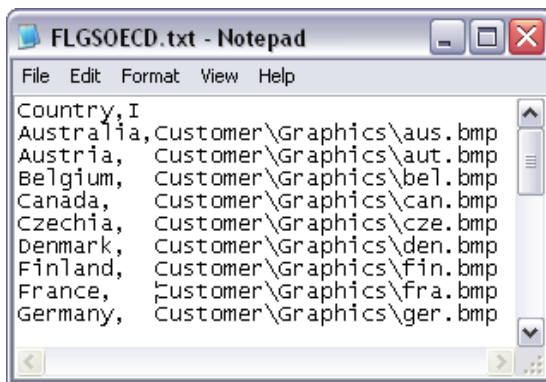



Figure 95. The information table defining the files linked to specific field values.

The statement for loading this table as an information table would be as follows:

```
Info Load Country, I from Flagsoecd.csv (ansi, txt,  
delimiter is ',', embedded labels);
```

When a list box or multi box item linked to information is selected, an information icon  appears beside the field name to show that information is available. Clicking the icon will show the information or load the application file. It is possible to turn off the information icon on the **Layout** page of the **List Box Properties** dialog.

If a text is entered in the second column, the text is shown in an internal text viewer.

To mark a new line in this text, the carriage return cannot be used. Instead, the symbol "\n" is used in the info file.

- If a name of an image file (e.g. extension bmp) is entered in the second column, the image is shown in an internal image viewer.
- If a name of a sound file (extension wav) is entered in the second column, the sound is played.
- If a name of an executable file is entered in the second column, the file is executed.
- If a name of any other file is entered in the second column, the associated program is used to open the file.
- If a URL is entered in the second column, e.g. an Internet address, the registered Internet browser is used to access the address.

An info file cannot contain the star symbol. A symbol defined as **OtherSymbol** (see page 491), however, is allowed.

Using **bundle info load**, you can bundle external data into a QlikView document. Read more about the syntax on page 205.

24.8 Data Cleansing

When loading data from different tables, you'll note that field values denoting the same thing are not always consistently named. Since this lack of consistency is not only annoying, but also hinders associations, you need to solve the problem. This can be done in an elegant way by creating a mapping table for the comparison of field values.

Mapping Tables

Tables loaded via **mapping load** or **mapping select** are treated differently from other tables. They will be stored in a separate area of the memory and

used only as mapping tables during script execution. After the script execution they will be automatically dropped.

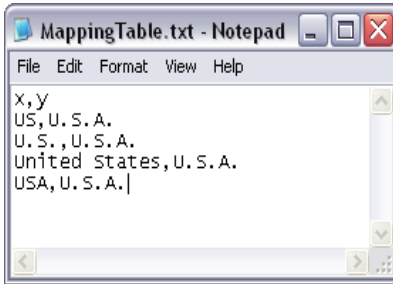


Figure 96. Example of a mapping table

Rules:

- A mapping table must have two columns, the first one containing the comparison values and the second the desired mapping values.
- The two columns must be named, but the names have no relevance in themselves. The column names have no connection to field names in regular input tables.

Using a Mapping Table

When loading several tables listing countries, you may find that one and the same country has several different names. In this example, the U.S.A. are listed as US, U.S., and United States.

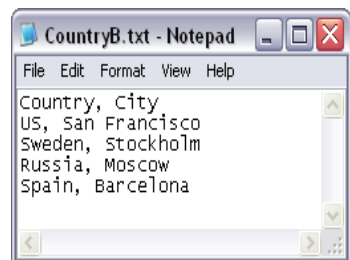
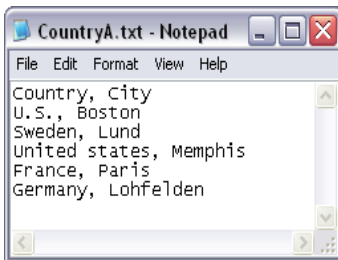


Figure 97.

To avoid the occurrence of three different records denoting the United States in the concatenated table, you can create a table similar to that in Figure 96 and load it as a mapping table.

The entire script should have the following appearance:

```

CountryMap:
Mapping Load x,y from MappingTable.txt
    (ansi, txt, delimiter is ',', embedded
    labels);
Map Country using CountryMap;
Load Country,City from CountryA.txt
    (ansi, txt, delimiter is ',', embedded labels);
Load Country, City from CountryB.txt
    (ansi, txt, delimiter is ',', embedded labels);

```

The **mapping** statement (for a detailed description of the syntax see page 318) loads the file *MappingTable.txt* as a mapping table with the label *CountryMap*.

The **map** statement (for a detailed description of the syntax see page 319) enables mapping of the field *Country* using the previously loaded mapping table *CountryMap*.

The **load** statements load the tables *CountryA* and *CountryB*. These tables, which will be concatenated due to the fact that they have the same set of fields (see the section on concatenation on page 467), include the field *Country*, whose field values will be compared with those of the first column of the mapping table. The field values *U.S.*, *United States* and *US* will be found and replaced by the values of the second column of the mapping table, i.e. *USA*.

The automatic mapping is done last in the chain of events that leads up to the field being stored in the QlikView table. For a typical **load** or **select** statement the order of events is roughly as follows:

- 1 Evaluation of expressions
- 2 Renaming of fields by as
- 3 Renaming of fields by alias
- 4 Qualification of table name, if applicable
- 5 Mapping of data if field name matches

This means that the mapping is not done every time a field name is encountered as part of an expression but rather when the value is stored under the field name in the QlikView table.

To disable mapping, use the **unmap** statement (see page 340).

For mapping on expression level, use the **applymap** function (see page 411).

For mapping on substring level, use the **mapsubstring** function (see “Mapping Functions” on page 411).

24.9 Wild Cards in the Data

It is also possible to use wild cards in the data. Two different wild cards exist: the star symbol, interpreted as all values of this field, and an optional symbol, interpreted as *all remaining values* of this field.

The Star Symbol

The star symbol is interpreted as all (listed) values of this field, i.e. a value listed elsewhere in this table. If used in one of the system fields (*USERID*, *PASSWORD* or *SERIAL*) in a table loaded in the access section of the script, it is interpreted as all (also not listed) possible values of this field.

The star symbol is not allowed in information files. Also, it cannot be used in key fields, i.e. fields used to join tables.

There is no star symbol available unless explicitly specified. For information on how to make a **star** statement in the script, see page 336.

OtherSymbol

In many cases you need a way to represent all other values in a table, i.e. all values that were not explicitly found in the loaded data. This is done with a special variable called **OtherSymbol**. To define the **OtherSymbol** to be treated as 'all other values', use the following syntax:

```
SET OTHERSYMBOL=<sym>;
```

before a **load/select** statement. <sym> may be any string.

The appearance of the defined symbol in an input table will cause QlikView to define it as all values not previously loaded in the field where it is found. Values found in the field after the appearance of the **OtherSymbol** will thus be disregarded.

In order to reset this functionality use:

```
SET OTHERSYMBOL=;
```

Example:

CUSTOMERS	
CustomerID	Name
1	ABC Inc.
2	XYZ Inc.
3	ACME Inc
+	Undefined

ORDERS	
CustomerID	OrderID
1	1234
3	1243
5	1248
7	1299

Insert the following statement in the script before the point where the first table above is loaded:

```
SET OTHERSYMBOL=+;
```

All references to *CustomerIDs* other than 1, 2 or 3, e.g. as when clicking on *OrderID 1299* will result in *Undefined* under *Name*.

Note **OtherSymbol** is not intended to be used for creating outer joins between tables!!

24.10 NULL Value Handling in QlikView

When no data can be produced for a certain field as a result of a database query and/or a join between tables, the result is normally NULL values.

The QlikView logic treats the following as real NULL values:

- NULL values returned from an ODBC connection
- NULL values created as a result of a forced concatenation of tables in the QlikView script
- NULL values created as a result of a **join** statement made in the QlikView script
- NULL values created as a result of the generation of field value combinations to be displayed in a table box or exported by an export button.

Normally it is impossible to use these NULL values for associations and selections. However, it is possible to manipulate NULL values at field level, using the **nullas-value** statement. See “NullAsNull” on page 322. Furthermore, NULL values from ODBC, can be manipulated using **nulldisplay** (see below).

Text files per definition cannot contain NULL values.

Associating/Selecting NULL Values from ODBC

It is possible to associate and/or select NULL values from an ODBC data source. For this purpose a script macro variable has been defined. By using the syntax:

```
SET NULLDISPLAY=<sym>;
```

the symbol <sym> will substitute all NULL values from the ODBC data source on the lowest level of data input. <sym> may be any string.

In order to reset this functionality to the default interpretation, use:

```
SET NULLDISPLAY=;
```

Note The use of **NULLDISPLAY** only affects data from an ODBC data source!

If you wish to have the QlikView logic with regard to NULL values from ODBC to interpret NULL values as an empty string, assign an empty string to the **NULLDISPLAY** variable in the following way (two single quotation marks without anything in between):

```
SET NULLDISPLAY=" ;
```

The assignment must be made before any **select** statement in the script. The empty string is then treated as any other value and thus makes the association and selection of NULL values possible.

Creating NULL Values from Text Files

It is possible to define a symbol, which when it occurs in a text file or an **inline** clause will be interpreted as a real NULL value. Use the following statement:

```
SET NULLINTERPRET=<sym>;
```

where <sym> is the symbol to be interpreted as NULL. <sym> may be any string.

In order to reset this functionality to the default interpretation, use:

```
SET NULLINTERPRET=;
```

Note The use of **NULLINTERPRET** only affects data from text files and **inline** clauses!

Propagation of NULL Values in Expressions

NULL values will propagate through an expression according to a few logical and quite reasonable rules.

Functions

The general rule is that functions return NULL when the parameters fall outside the range for which the function is defined.

Examples:

asin(2)	returns	NULL
log(-5)	returns	NULL
round(A,0)	returns	NULL

As a result of the above follows that functions generally return NULL when any of the parameters necessary for the evaluation are NULL.

Examples:

sin(NULL)	returns	NULL
chr(NULL)	returns	NULL
if(NULL, A, B)	returns	B
if(TRUE, NULL, A)	returns	NULL
if(TRUE, A, NULL)	returns	A

The exception to the second rule are logical functions testing for type.

Examples:

isnull(NULL)	returns	TRUE (-1)
isnum(NULL)	returns	FALSE (0)

Arithmetic and String Operators

If NULL is encountered on any side of these operators NULL is returned, except in the case of string concatenation.

Examples:

A + NULL	returns	NULL
A - NULL	returns	NULL
A / NULL	returns	NULL
A * NULL	returns	NULL
NULL / A	returns	NULL
NULL * A	returns	NULL
0 / NULL	returns	NULL

0 * NULL	returns	NULL
A&NULL	returns	A

Relational Operators

If NULL is encountered on any side of relational operators, special rules apply.

Examples:

NULL rel.op. NULL	returns	NULL
A <> NULL	returns	TRUE (-1)
A < NULL	returns	FALSE (0)
A <= NULL	returns	FALSE (0)
A = NULL	returns	FALSE (0)
A >= NULL	returns	FALSE (0)
A > NULL	returns	FALSE (0)

24.11 Character Sets

By default QlikView assumes that the character set used in Windows is **ansi**. However, sometimes files with other character sets are used, e.g. when text files are transferred from other computers. The most common character set, apart from **ansi**, is the **oem** character set. This is used in DOS, OS/2 etc. By using the **oem** file specifier in the **load** statement, QlikView will use the **oem** character set to interpret the data. QlikView can also recognize and use the MacOS-specific character set **mac**. See also the script syntax on page 315.



25 QVD FILES

25.1 About QVD Files

A QVD (QlikView Data) file is a file containing a table of data exported from QlikView. QVD is a native QlikView format. It can only be written to and read from QlikView. The file format is optimized for speed when reading data from a QlikView script but it is also very compact. Reading data from a QVD file is typically 10-100 times faster than reading from other data sources.

25.2 The QVD file Format

A QVD file is an attempt to strike a compromise between excellent QlikView performance both reading and writing the files and a compact representation. A QVD file contains exactly one table. Conceptually it is quite similar to any typed file (e.g. csv, dif, biff or fix). A QVD file consists of three parts:

- 1 A well formed XML header (in UTF-8 char set) describing the fields in the table, the layout of the subsequent information and some other meta-data.
- 2 Symbol tables in a byte stuffed format.
- 3 Actual table data in a bit-stuffed format.

25.3 What are QVD files good for?

QVD files can be used for many purposes. At least four major uses can be easily identified. In many cases two or more of them will be applicable at the same time. They are:

Increasing Load Speed

By buffering non-changing or slowly-changing parts of input data in QVD files, script execution can become considerably faster for large data sets. For large data sets it will thus be easier to meet reload time-window limitations. When developing applications it is often necessary to run the script repeatedly. By using QVD buffering in such situations repeated waiting times can be reduced significantly even if the data set is not that large.

Decreasing Load on Database Servers

By buffering non-changing or slowly-changing parts of input data in QVD files, the amount of data fetched from external data sources can be greatly

reduced. This reduces load on external databases and network traffic. When several QlikView scripts share the same data it is only necessary to load it once from the source database. The other applications can make use of the data from a QVD file.

Consolidating Data from Multiple QlikView Applications

Consolidation of data from multiple QlikView applications is possible with the help of QVD files. With the Binary script statement you can only load data from only one single QlikView application into another. With QVD files a QlikView script can combine data from any number of QlikView applications. This opens up possibilities e.g. for applications consolidating similar data from different business units etc.

Incremental Load

In many common cases the QVD functionality can be used to facilitate incremental load, i.e. only loading new records from a growing database.

25.4 How Fast Is It?

Reading data from a QVD file will always be faster than reading from other sources. A factor of ten to a hundred times faster will not be uncommon. The exact speed in terms of record/second throughput will depend on a number of factors, the most important being:

- number of fields in record
- type of data in fields (field width, number of distinct values etc.)
- speed of hard disk
- speed of computer CPU

QVD files can be read in two modes, standard (fast) and super-fast. Which mode is used is determined automatically by the QlikView script engine. Super-fast mode can be used only when all fields or a subset thereof are read without any transformations (formulas acting upon the fields). Renaming of fields is allowed. Whenever transformations are applied the standard (fast) mode will be used.

25.5 Creation of QVD Files

QVD files can be created in three manners:

- 1 Explicitly created and named from script by means of the **store** command. You simply state in the script that you want a previously read table or part thereof to be exported to an explicitly named filename at a location that you choose. See technical details below.
- 2 Automatically created and maintained from script. By preceding a **load** or **select** statement with the new **buffer** prefix QlikView will automatically create a QVD file which later, if certain conditions are met will be used instead of the original data source when reloading data. The QVD file will have a cryptic name based on a hash of the load/select statement and normally reside in the Windows Application data folder. See technical details below.
- 3 Explicitly named and created from layout or via Automation. Data exported from the QlikView layout via GUI commands or Automation macros. In the GUI you will find QVD as one of the possible export formats under the **Export...** command, found on the object menu of most sheet objects.

The resulting QVD file has the same attributes (such as size and reading speed) regardless of which method is used.

Manual Creation from Script

A QVD file can be created by an **store** statement in the script. This is the statement that will create an explicitly named QVD file. The syntax is:

```
store [*fieldlist from] table into filename;
```

The <table> is a script labeled, already loaded, table. The <file_name> is interpreted similar to names in **load** statements, i.e. the **directory** statements apply. Fields in the fields list may be renamed using standard **as** syntax.

Examples:

```
STORE mytable INTO xyz.qvd;
STORE * FROM mytable INTO xyz.qvd;
STORE Name, RegNo FROM mytable INTO xyz.qvd;
STORE Name as a, RegNo as b FROM mytable INTO xyz.qvd;
```

The two first statements have identical function.

Buffering - Automatic Creation from Script

QVD files can be created and maintained automatically via the **buffer** prefix. This prefix can be used on most **load** and or **select** statements in script. It indicates that a QVD file is used to cache/buffer the result of the statement.

Some limitations exist, the most notable is that there must be either a file load or a **select** statement in “the bottom”.

The name of the QVD file is a *calculated* name (a 160-bit hash of statement and other discriminating info, as hex) and is typically stored in the APP-DATA folder (C:\Document and Settings\%user%\Application Data\Qlik-Tech\QlikView).

The syntax of the prefix is:

```
BUFFER [(option [,option])] LOAD ...
```

or

```
BUFFER [(option [,option])] SELECT ...
```

where an option is either of the following:

incremental

this enables the ability to read only part of an underlying file. Previous size of the file is stored in the XML header in the QVD file. This is particularly useful with log files. All records loaded at a previous occasion are read from the QVD file whereas the following new records are read from the original source and finally an updated QVD file is created.

stale (after) amount [(days | hours)]

This is typically used with DB sources where there is no simple timestamp on the original data. Instead one specifies how old the QVD snapshot can be to be used.

Examples:

```
BUFFER SELECT * FROM MyTable;  
BUFFER (stale after 7 days) SELECT * FROM MyTable;  
BUFFER (incremental) LOAD * FROM MyLog.log;
```

Manual Creation from Layout

Export from sheet objects into QVD files is possible from the existing **Export** commands and via Automation.

25.6 Reading data from QVD files

Use of a QVD file as explicit data source

QVD files can be referenced by a load statement in the QlikView script just like any other type of text file (csv, fix, dif, biff etc.). The standard **Table Files Wizard** works with QVD files just like any other table file.

```
LOAD * FROM xyz.qvd (qvd);  
LOAD Name, RegNo FROM xyz.qvd (qvd);  
LOAD Name as a, RegNo as b FROM xyz.qvd (qvd);
```

When no transformations are applied on the fields read (apart from renaming fields) the super-fast reading mode will be used.

Automatic use of buffered QVD files

When using the buffer prefix on load or select statements, no explicit statements for reading are necessary. QlikView will determine to which extent to use data from the QVD file or acquire data via the original load or select statement.

New script functions

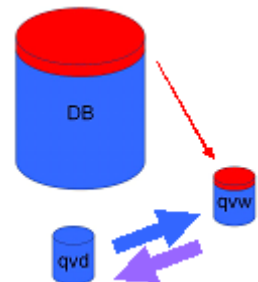
A number of new script functions have been added for access to the data found in the XML header of a QVD file. These functions are described under “File Functions” on page 421 in this book.

25.7 Using QVD files for incremental load

Incremental load is a very common task in relation to data bases. It is defined as loading only the new or changed records from the database. The rest should already be available, one way or another.

With the help of QVD files it is possible to perform incremental load in most cases. The basic process goes like follows:

- 1 Load new data from Database table (slow, but few records)
- 2 Load old data from QVD file (many records, but fast)
- 3 Create new QVD file
- 4 Repeat procedure for each table



The actual complexity of the solution depends on the conditions of the source database, but can be broken down into a few basic cases, as outlined below.

- 1 Append only (Logfiles)
- 2 Insert only (No Update or Delete)
- 3 Insert and Update (No Delete)
- 4 Insert, Update and Delete.

Below you will find outlined solutions for each of these cases.

Case 1: Append only

The simplest case is the one of log files, which are only appended to. The conditions are as follows:

- The database must be a log file contained in a text file (no ODBC/OLEDB)
- QlikView keeps track of how many records have been read previously and loads only records added in the end of the file.



Script example:

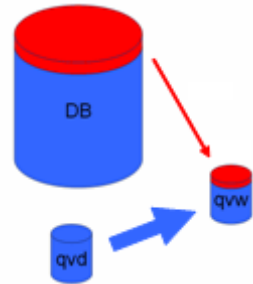
```
Buffer (Incremental) Load * From LogFile.txt (ansi,
txt, delimiter is '\t', embedded labels);
```

Case 2: Insert only. (No Update or Delete)

If the data resides in a database other than a simple log file the case 1 approach will not work.

The problem can however still be solved with a small amount of extra work. The conditions are as follows:

- The data source can be any database
- QlikView loads records inserted in the database after the last script execution
- A field ModificationDate (or similar) is required for QlikView to know which records are new.



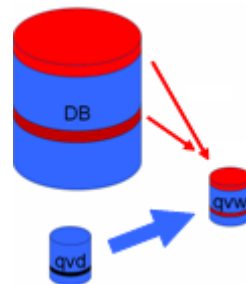
Script example:

```
QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#
    AND ModificationTime < #$(BeginningThisExecTime)#;
Concatenate
LOAD PrimaryKey, X, Y FROM File.QVD;
STORE QV_Table INTO File.QVD;
```

Case 3: Insert and Update. (No Delete)

The next case is when data in previously loaded records may have been changed between script executions. The conditions are as follows:

- The data source can be any database
- QlikView loads records inserted into the database or updated in the database after the last script execution
- A field ModificationDate (or similar) is required for QlikView to know which records are new.
- A primary key field is required for QlikView to sort out updated records from the QVD file.
- This solution will make the reading of the QVD file to be made in the standard mode rather than the super-fast mode. The end result will however still be considerably faster than if the entire database had to be read.



Script example:

```

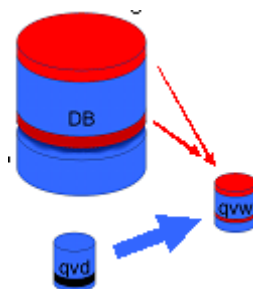
QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#;
Concatenate
LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT Exists(PrimaryKey);
STORE QV_Table INTO File.QVD;

```

Case 4: Insert, Update and Delete

The most difficult case to handle is when records are actually deleted from the source database between script executions. The conditions are as follows:

- The data source can be any database
- QlikView loads records inserted into the database or updated in the database after the last script execution
- QlikView removes records deleted from the database after the last script execution



-
- A field ModificationDate (or similar) is required for QlikView to know which records are new.
 - A primary key field is required for QlikView to sort out updated records from the QVD file.
 - This solution will make the reading of the QVD file to be made in the standard mode rather than the super-fast mode. The end result will however still be considerably faster than if the entire database had to be read.

Script example:

```
Let ThisExecTime = Now();
QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#
      AND ModificationTime < #$(ThisExecTime)#;

Concatenate LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT EXISTS(PriamryKey);

Inner Join SQL SELECT PrimaryKey FROM DB_TABLE;

If ScriptErrorCount = 0 then
    STORE QV_Table INTO File.QVD;
    Let LastExecTime = ThisExecTime;
End If
```


26 SECURITY

It is important that information is distributed only to those who have rights to see it. Since QlikView makes the previously cumbersome process of retrieving information a very simple task, it is obvious that a mechanism that handles security issues is necessary. Such a mechanism can be set up in two different ways: It can either be built into the QlikView document script, or it can be set up through the use of QlikView Publisher.

Further, it may also be important to lock the layout for normal users, or prevent them from editing or running the script, etc. This can also be done through the access restriction built into the QlikView file.

26.1 Authentication and Authorization

Authentication is any process by which you verify that someone is who they claim they are. QlikView can either let the Windows operating system do the authentication, or prompt for a User ID and Password (different from the Windows User ID and Password) or use the QlikView license key as a simple authentication method.

Authorization is finding out if the person, once identified, is permitted to have the resource. QlikView can either let the Windows operating system do the authorization or do the authorization itself. For the latter, a security table must be built into the script.

26.2 Security using the QlikView Publisher

If the QlikView Publisher is set up to handle security, then each QlikView file will be split up into several files, each containing the data pertaining to the relevant user or user group. These files will be stored in folders with the correct OS security settings, i.e. QlikView lets the operating system handle Authentication and Authorization.

There is, however, *no security built into the file itself*, so there is no protection on a downloaded file.

The file sizes will usually be smaller, since one single file will be split into several and the user only opens the file with his own data. However, this also means that a QlikView Server can potentially use more memory than if all data are kept in one file, since several files containing the same data sometimes will be loaded.

For further information, see the QlikView Publisher documentation.

26.3 Security using the Section Access in the QlikView script

If the Section Access in the QlikView script is set up to handle security, then one single file can be made to hold the data for a number of users or user groups. QlikView will use the information in the Section Access for Authentication and Authorization and dynamically reduce the data, so that the user only sees his own data.

The security is built into the file itself, so also a downloaded file is to some extent protected. However, if the security demands are high, downloads of files and offline use should be prevented. The files should be published by the QlikView Server only.

Since all data are kept in one file, the size of this file can potentially be very large.

All information below refers to the security method of using Section Access in the QlikView script.

26.4 Sections in the Script

Access control is managed via one or several security tables loaded in the same way as QlikView normally loads data. It is thus possible to store these tables in a normal database. The script statements managing the security tables are given within the access section, which in the script is initiated by the statement **section access**, see page 329.

If an access section is defined in the script, the part of the script loading the "normal" data must be put in a different section, initiated by the statement **section application**.

Example:

```
Section Access;  
Load * inline  
    [ACCESS,USERID,PASSWORD  
    ADMIN, A,X  
    USER,U,Y ];  
Section Application;  
Load... ... from... ...
```

26.5 Access Levels in Section Access

Access to QlikView documents can be authorized to specified users or groups of users. In the security table, users can be assigned the access levels ADMIN or USER. If no access level is assigned, the user cannot open the QlikView document. For clarity, it is often useful to use a third access level, e.g. NONE, which of course will be interpreted as "no access".

A person with ADMIN access can change everything in the document. Using the **Security** page in the **Document Properties** (“Security” on page 42) and **Sheet Properties** (“Security” on page 85) dialogs, a person with ADMIN access can limit the users’ possibilities of modifying the document. A person with USER privileges cannot access the Security pages.

26.6 Section Access system fields

The access levels are assigned to users in one or several tables loaded within the **section access**. These tables can contain several different user-specific system fields, typically *USERID* and *PASSWORD*, and the field defining the access level, *ACCESS*. All Section Access system fields will be used for authentication or authorization. The full set of **section access** system fields are described below. Other fields like e.g. *GROUP* or *ORGANISATION* may be added to facilitate the administration, but QlikView does not treat these fields in any special way.

None, all, or any combination of the security fields may be loaded in the access section. However, if the *ACCESS* field is not loaded, all the users will have ADMIN access to the document and the **access** section will really not be meaningful.

It is thus not necessary to use *USERID* – an authorization can be made using other fields, e.g. serial number only.

<i>ACCESS</i>	A field that defines what access the corresponding user should have.
<i>USERID</i>	A field that should contain an accepted user ID. QlikView will prompt for a User ID and compare to the value in this field. This user ID is not the same as the Windows user ID.
<i>PASSWORD</i>	A field that should contain an accepted password. QlikView will prompt for a Password and compare to the value in this field. This password is not the same as the Windows password.
<i>SERIAL</i>	A field that should contain a number corresponding to the QlikView serial number. Example: 4900 2394 7113 7304. QlikView will check the serial number of the user and compare it to the value in this field.
<i>NTNAME</i>	A field that should contain a string corresponding to a Windows NT Domain user name or group name. QlikView will fetch the logon information from the OS and compare it to the value in this field.
<i>NTDOMAINSID</i>	A field that should contain a string corresponding to a Windows NT Domain SID.

Example: S-1-5-21-125976590-467238106-1092489882

QlikView will fetch the logon information from the OS and compare it to the value in this field.

NTSID

A field that should contain a Windows NT SID.

Example: S-1-5-21-125976590-467238106-1092489882-1378

QlikView will fetch the logon information from the OS and compare it to the value in this field.

OMIT

A field that should contain the field that should be omitted for this specific user. Wildcards may be used and the field may be empty. A facile way of doing this is to use a subfield.

QlikView will compare the QlikView serial number with the field *SERIAL*, the Windows NT User name and groups with *NTNAME*, the Windows NT Domain SID with *NTDOMAINSID* and the Windows NT SID with *NTSID*. It will further prompt for User ID and Password and compare these with the fields *USERID* and *PASSWORD*.

If the found combination of user ID, password and environment properties is also found in the **section access** table, then the document is opened with the corresponding access level. If not, QlikView will deny the user access to the document. If the User ID and/or the Password are not entered correctly within three attempts the entire log-on procedure must be repeated.

Since the same internal logic that is the hallmark of QlikView is used also in the access section, the security fields may be put in different tables. (It is thus possible for a system manager to make a QlikView document out of the security tables. In this case a correct serial number, password etc. is simulated by a click on the corresponding field value.)

In the logon procedure, QlikView will first check *SERIAL*, *NTNAME*, *NTDOMAINSID* and *NTSID* to see if this information is enough to grant the user access to the document. If so, QlikView will open the document without prompting for User ID and Password.

If only some of the access fields are loaded, the appropriate of the above requirements are used.

All the fields listed in **Load** or **Select** statements in the **section access** must be written in UPPER CASE. Any field name containing lower case letters in the database will be converted to upper case before being read by the Load or Select statement. However the user ID and the password entered by the end-user opening the QlikView documents are case insensitive.

Note To generate access tables in inline statements use the Access Restriction Table Wizard, see page 244.

Example 1:

Only serial number is checked. One specific computer gets ADMIN access. Everyone else gets USER access. Note that a star can be used to mark “any serial number”. For this, however, the statement “Star is *;” must precede the loading of this table.

ACCESS	SERIAL
ADMIN	4900 2394 7113 7304
USER	*

Example 2:

The administrator and the server on which QlikView runs as a batch job get ADMIN access. Everyone else in the Domain gets USER access when entering “USER” as user ID and password.

ACCESS	SERIAL	NTDOMAINSID	USERID	PASSWORD
ADMIN	*	S-1-5-21-125976590-467238106-1092489882	ADMIN	ADMIN
ADMIN	4900 2394 7113 7304	*	*	*
USER	*	S-1-5-21-125976590-467238106-1092489882	USER	USER

26.7 Restrictions on QlikView functionality

The controls found on the **Security** page of the **Document Properties** dialog (see page 42 in Book II) and the **Security** page of the **Sheet Properties** dialog (see page 85 in Book II) makes it possible to disallow the access to certain menu items and prohibit changes in the layout. If these settings are to be used as a truly protective measure, it is important that the document users are logged in as USER. Anyone logged in as ADMIN can change the security settings at any time.

A user that has opened the document with USER rights does not have the **Security** pages in the Properties dialogs.

26.8 Dynamic Data Reduction

QlikView and QlikView Server support a feature by which some of the data in a document can be hidden from the user based on the **section access** login.

First of all, fields (columns) can be hidden by the use of the system field **OMIT**.

Secondly, records (rows) can be hidden by linking the **Section Access** data with the real data: The selection of values to be shown/excluded is controlled by means of having one or more fields with common names in **section access** and **section application**. After user login QlikView will attempt to copy the selections in fields in **section access** to any fields in **section application** with exactly the same field names (the field names must be written in UPPER CASE). After the selections have been made, QlikView will permanently hide all data excluded by these selections from the user.

In order for this procedure to take place, the option **Initial Data Reduction Based on Section Access** on the **Opening** page of the **Document Properties** dialog must be selected. If this feature is used in documents that are to be distributed by other means than via QlikView Server, the option **Prohibit Binary Load** on the same page of the Document Properties must be selected in order to maintain data protection.

Note All field names used in the transfer described above and all field values in these fields must be upper case, since all field names and field values are by default converted to upper case in **section access**.

Example:

```
section access;
load * inline [
    ACCESS, USERID, REDUCTION, OMIT
    ADMIN, ADMIN, *,
    USER, A, 1
    USER, B, 2, NUM
    USER, C, 3, ALPHA
];

section application;
T1:
load *,
    NUM AS REDUCTION;
load
    Chr( RecNo()+ord('A')-1) AS ALPHA,
    RechNo() AS NUM
AUTOGENERATE 3;
```

The field REDUCTION (upper case) now exists in both **section access** and **section application** (all field values are also upper case). The two fields would normally be totally different and separated, but if the **Initial Data Reduction Based on Section Access** option has been selected, they will link and reduce the number of records displayed to the user.

The field OMIT in **section access** defines the fields that should be hidden from the user.

The result will be as follows:

User A can see all fields, but only those records connected to REDUCTION=1.

User B can see all fields except NUM, and only those records connected to REDUCTION=2.

User C can see all fields except ALPHA, and and only those records connected to REDUCTION=3.

26.9 Inherited Access Restrictions

A **binary** load will cause the access restrictions to be inherited by the new QlikView document. A person with ADMIN rights to this new document may change the access rights of this new document by adding a new **access** section. A person with USER rights can execute the script and change the script, thus adding own data to the binary loaded file. A person with USER rights cannot change the access rights. This makes it possible for a database administrator to control the user access also to binary loaded QlikView documents.

26.10 Encryption

The communication between a QlikView Server and a QlikView Windows client is encrypted. If, however, the Java client or the AJAX client is used, the communication is not encrypted.

In addition, all QlikView documents are scrambled which makes the information unreadable with viewers, debuggers etc.



INDEX

Symbols

! (forced exclusion)	137
? (single character wildcard)	131
@ (field number specifier)	309
* (load all fields)	309
* (star symbol)	491
& (string concatenation)	356

A

Access Restriction	506–511
Acos	394
Add statement	284
AddMonths	432
Age	448
Aggregation functions in script expressions	359
Alerts	183–189
Alias statement	285
Alt	416
an alert	183
And	357
And-mode	137
Applycodepage	411
Applymap	411
As clause	310
Asin	394
Associating tables	463–466
Atan	394
Atan2	395
Attribute	421
Autogenerate	311
Autonumber	404
Autonumberhash128	404
Autonumberhash256	405
Avg	364

B

Basic aggregation functions in script	
firstsortedvalue	361
max	360
min	360
mode	360
only	360

Batch Execution	51
biff	315
Binary statement	285
Bit operators in script	
bitand	358
bitnot	358
bitor	359
bitxor	359
Bitand	358
Bitcount	388
Bitnot	358
Bitor	359
Bitxor	359
BlackAndSchole	398
Bookmarks	149–159
Bookmarks menu	69
Buffer statement	286
Bundle keyword	287

C

Call statement	288
Capitalize	407
CD	
variable in script	344
Ceil	385
Character set	315
Chi2test_chi2	373
Chi2test_df	372
Chi2test_p	372
Chidist	395
Chiinv	395
Chr	405
Circular references	466
avoiding	145
Class	417
Combin	387
Command line	
switches	45
syntax	45
Comment in script	317, 325
Compatibility	
file	55
Compatibility Issues	29
ComputerName	419

Concat	362
Concatenate statement	289
Concatenation	
automatic	467
forced	468
Conditional functions	
in script expressions	415
Conditional functions in script	
alt	416
class	417
if	415
match	416
mixmatch	416
pick	416
wildmatch	417
wildmatch5	417
Configuring	
ODBC data sources	41
toolbar	84
Connect statement	290
ConnectString	423
ConvertToLocalTime	430
Correl	365
Cos	394
Cosh	395
Count	362
Counter aggregation functions in script	
count	362
MissingCount	364
NullCount	363
NumericCount	363
TextCount	363
Counter functions	
in script expressions	402
Counter functions in script	
autonumber	404
autonumberhash128	404
autonumberhash256	405
fieldvaluecount	405
IterNo	403
RecNo	402
RowNo	402
Create	
QVD files	498
Creating	183
Cross tables	476
Crosstable statement	291
Current selections	141

D

Data	
cleansing	488
source	231
Data Link Properties	231
Date	451, 455
Date and time functions in script	
AddMonths	432
age	448
ConvertToLocalTime	430
day	426
DayEnd	441
daylightsaving	448
DayName	442
DayNumberOfQuarter	432
DayNumberOfYear	432
DayStart	441
firstworkdate	449
hour	429
InDay	447
InDayToTime	447
InLunarWeek	446
InLunarWeekToDate	446
InMonth	444
InMonths	445
InMonthsToDate	445
InMonthToDate	444
InQuarter	443
InQuarterToDate	444
InWeek	445
InWeekToDate	446
InYear	443
InYearToDate	443
lastworkdate	449
LocalTime	430
Lunarweek	439
LunarweekEnd	440
LunarWeekName	440
MakeDate	428
MakeTime	428
MakeWeekDate	428
minute	429
month	427
MonthEnd	436
MonthName	436
MonthsEnd	437
MonthsName	437
MonthsStart	436
MonthStart	435
networkdays	448

now	429	ReportNumber	426
QuarterEnd	435	Dollar-sign Expansions	350
QuarterName	435	Drop field statement	293
QuarterStart	434	Drop table statement	293
second	429	Dual	457
setdateyear	442	E	
setdateyearmonth	442	e, base of natural logarithms	402
today	430	Edit menu	62
UTC	448	Error Variables	349
week	426	ErrorMode	
weekday	427	variable in script	349
WeekEnd	438	Evaluate	410
WeekName	439	Even	387
WeekStart	438	Exclusion	137
weekyear	427	Execute statement	294
year	427	Exists	414
YearEnd	433	Exit script statement	294
YearName	434	Exp	394
YearStart	433	Exponential and Logarithmic functions in script	
yeartodate	431	exp	394
Date functions		log	394
in script expressions	426	log10	394
DateFormat		pow	394
variable in script	348	sqr	394
Day	426	sqrt	394
DayEnd	441	Exponential functions	
Daylightsaving	448	in script expressions	394
DayName	442	Expression Syntax	355
DayNames		Expressions	
variable in script	348	in script	355–459
DayNumberOfQuarter	432	F	
DayNumberOfYear	432	Fabs	386
DayStart	441	Fact	387
Debugger	277	False	402
DecimalSep		Fdist	397
variable in script	347	Fieldindex	415
delimiter in text file	316	FieldName	425
Design Grid	65	FieldNr	425
dif	315	Fieldvalue	414
Directory statement	291	Fieldvaluecount	405
Disconnect statement	292	File	
Distinct predicate	309	compatibility	55
Div	384	menu	57
Do..loop statement	292	File functions	
Document Functions	426	in script expressions	421
Document functions in script		File functions in script	
NoOfReports	426		
ReportComment	426		
ReportID	426		
ReportName	426		

attribute	421	For each..next statement	296
ConnectString	423	For..next statement	295
FileBasename	421	Force statement	298
FileDir	421	Formatting functions	
FileExtension	422	in script expressions	453
FileName	422	Formatting functions in script	
FilePath	422	date	455
FileSize	422	dual	457
FileTime	423	interval	457
GetFolderPath	423	money	454
QvdCreateTime	423	num	454
QvdFieldName	424	time	455
QvdNoOfFields	424	timestamp	456
QvdNoOfRecords	423	Frac	386
QvdTableName	424	Fractile	364
File wizard	245–279	From clause	310
FileBasename	421	FTP file browser dialog	239
FileDir	421	Function in script expressions	
FileExtension	422	Range	388
FileName	422	Functions in script expressions	
FilePath	422	aggregation	359
FileSize	422	conditional	415
FileTime	423	counter	402
financial aggregation function in script		date and time	426
irr	371	exponential	394
npv	371	file	421
xirr	371	financial	397
xnpv	372	formatting	453
Financial functions		hyperbolic	394
in script expressions	397	interpretation	449
Financial functions in script		inter-record	412
BlackAndSchole	398	logarithmic	394
fv	399	logical	417
nper	399	mapping	411
pmt	400	mathematical constants	402
pv	400	null	418
rate	401	numeric	384
Find/Replace Dialog	226	statistical distribution	395
Findoneof	410	string	405
Findv	397	system	418
First statement	295	table	424
Firstsortedvalue	361	trigonometric	394
FirstValue	362	Fuzzy Search	131
Firstworkdate	449	Fv	399
fixed record length	315		
Floor	385		
Floppy		G	
variable in script	344	Generic	
Fmod	385	databases	475
Follows	358	statement	299
		GetFolderPath	423

GetRegistryString	419
Group by clause	311

H

Hash128	411
Hash160	411
Hash256	411
Having clause	331
Header size in text file	317
Help menu	74
Hidden script	227
HidePrefix	
variable in script	345
HideSuffix	
variable in script	345
Hierarchies	481–483
Hierarchy statement	299
HierarchyBelongsTo statement	300
Hour	429
Hyperbolic functions	
in script expressions	394

I

If	415
If..then..elseif..else..end if statement ..	301
Image_size statement	302
Include	
variable in script	345
InDay	447
InDayToTime	447
Index	406
Info statement	302
Inline clause	311
Inline Data Wizard	241
InLunarweek	446
InLunarweekToDate	446
InMonth	444
InMonths	445
InMonthsToDate	445
InMonthToDate	444
Inner keyword	303
Input	419
Inputfield statement	304
InQuarter	443
InQuarterToDate	444
Installing	
ODBC	39–42
OLE DB	39–42
QlikView	35–38

Interface Language	94
Interpretation functions	
in script expressions	449
Inter-record function in script	
exists	414
fieldindex	415
fieldvalue	414
lookup	415
peek	413
previous	413
Inter-record functions	
in script expressions	412
Interval	452, 457
IntervalMatch	477
IntervalMatch statement	304
InWeek	445
InWeekToDate	446
InYear	443
InYearToDate	443
Irr	371
IsNull	418
IsNum	417
IsPartialReload	417
IsText	418
IterNo	403

J

Join and Keep	469
Join statement	306
Joining tables	469–470

K

Keep statement	306
Keep statement in script	470
KeepChar	407
Keywords	281–342
Kurtosis	365

L

Label	
field	316
table	342
LastValue	362
Lastworkdate	449
Layout	66
LEF file	37
Left	405
Left keyword	307
Len	405

Let statement	308
License Enabler File	38
Linest_b	366
Linest_df	369
Linest_f	369
Linest_m	366
Linest_r2	367
Linest_seb	368
Linest_sem	367
Linest_sey	368
Linest_ssreg	370
Linest_ssresid	370
Linking information	487–488
Load statement	309
LocalTime	430
Locations page	120
Locking selections	138
Log	394
Log10	394
Logarithmic functions in script expressions	394
Logical functions in script expressions	417
Logical functions in script	
IsNum	417
IsPartialReload	417
IsText	418
Logical operators in script	
and	357
not	357
or	357
xor	357
Logical tables	462
LongDaynames variable in script	348
LongMonthNames variable in script	348
Lookup	415
Loops	466
Loosely Coupled Tables	144
Loosen Table statement	318
Lower	407
Ltrim	407
Lunarweek	439
LunarweekEnd	440
LunarweekName	440

M

MakeDate	428
----------------	-----

MakeTime	428
MakeWeekDate	428
Map..using statement	319
Mapping statement	318
Mapping functions in script	
applymap	411
mapsubstring	412
Mapping functions in script expressions ... 411	
Mapsubstring	412
Match	416
Mathematical constants in script expressions	402
Mathematical Constants and Parameter- free functions in script	
e, base of natural logarithms	402
false	402
pi	402
rand	402
true	402
Max	360
MaxString	361
Median	364
Menu commands	
Bookmarks menu	69
Edit menu	62
File menu	57
Help menu	74
Layout menu	66
Object menu	72
Reports menu	70
Selections menu	65
Settings menu	68
Tools menu	71
View menu	64
Window menu	72
Mid	406
Min	360
MinString	361
Minute	429
MissingCount	364
Mixmatch	416
Mod	385
Mode	360
Money	450, 454
MoneyDecimalSep variable in script	347
MoneyFormat variable in script	347

MoneyThousandSep	
variable in script	347
Month	427
MonthEnd	436
MonthName	436
MonthNames	
variable in script	348
MonthsEnd	437
MonthsName	437
MonthStart	435, 436
MsgBox	419
Multiple selection	130

N

Naming	
tables	342
Network distribution of QlikView	37
Networkdays	448
Noconcatenate statement	321
NoOfFields	425
NoOfReports	426
NoOfRows	424
NoOfTables	425
Normdist	396
Norminv	396
Not	357
Notational conventions	15
not-selection	137
Now	429
Nper	399
Npv	371
Null	418
Null functions	
in script expressions	418
Null functions in script	
IsNull	418
Null	418
NULL value handling	492–495
NullAsNull statement	322
NullAsValue statement	321
NullCount	363
NullDisplay variable in script	346
NullInterpret variable in script	346
NullValue	
variable in script	346
Num	449, 454
Number interpretation functions in script	
date	451
interval	452
money	450

num	449
text	453
time	451
timestamp	452
Number interpretation variables	347
Numeric functions	
in script expressions	384
Numeric functions in script	
bitcount	388
ceil	385
combin	387
div	384
even	387
fabs	386
fact	387
floor	385
fmod	385
frac	386
mod	385
odd	388
permut	387
round	386
sign	386
Numeric search	132
NumericCount	363

O

Object menu	72
ODBC	39–42
data sources	41
Odd	388
OLE DB	39–42
Only	360
OpenUrlTimeout variable in script	346
Operators in script expressions	
bit	358
logical	357
numeric	356
relational	357
string	356
Or	357
Ord	405
Order by clause	331
Osuser	418
OtherSymbol	491
variable in script	347
Outer keyword	323

P	
Peek	413
Permut	387
Pi	402
Pick	416
Pmt	400
Pow	394
Precedes	358
Preceding Load	235
Previous	413
PurgeChar	408
Pv	400
Q	
QlikView	
documents	55
versions	55
QlikViewversion	418
Qualify statement	323
QuarterEnd	435
QuarterName	435
QuarterStart	434
Quotation marks	353
QvdCreateTime	423
QvdFieldName	424
QvdNoOfFields	424
QvdNoOfRecords	423
QvdTableName	424
QVPath	
variable in script	344
QVRoot	
variable in script	344
QVuser	418
QVWorkPath	
variable in script	344
QVWorkRootPath	
variable in script	344
R	
Rand	402
Range functions in script	
rangeavg	388
rangecorrel	389
rangelcount	389
rangefractile	390
rangeirr	392
rangekurtosis	390
rangemax	389
rangemaxstring	392
rangemin	389
rangeminstring	391
rangemissingcount	391
rangemode	391
rangenvp	392
rangemnullcount	391
rangenumericcount	390
rangeonly	391
rangeskew	390
rangestdev	390
rangesum	388
rangetextcount	390
rangexirr	393
rangexnpv	393
Rate	401
RecNo	402
Record length in text file	317
Relational operators in script	
follows	358
precedes	358
ReloadTime	419
Rem statement	325
Rename Field statement	325
Renaming fields	466
Repeat	407
Replace	408
Replace statement	326
ReportComment	426

ReportID	426	Sign	386
ReportName	426	Sin	394
ReportNumber	426	Sinh	395
Reports	161–181	Skew	365
Reports menu	70	Sleep	
Resident clause	311	statement	333
Right	406	Special Variables	344
Right keyword	328	SQL statement	333
Round	386	SQLCOLUMNS statement	333
RowNo	402	SQLTABLES statement	334
Rtrim	407	SQLTYPES statement	335
S		Sqr	394
Sample statement	328	Sqrt	394
Script		Star	
Edit Script dialog box	217	statement	336
expressions	355–459	symbol	491
keywords	281–342	Start Page	46
quotation marks	353	Starting QlikView	45
statements	281–342	from command line	45
syntax	281–354	Statements	281–342
variables	343–350	Statistical	372
ScriptError		Statistical aggregation functions in script	
variable in script	349	avg	364
ScriptErrorCount		correl	365
variable in script	350	fractile	364
ScriptErrorDetailed		kurtosis	365
variable in script	350	linest_b	366
ScriptErrorList		linest_df	369
variable in script	350	linest_f	369
Second	429	linest_m	366
Section statement	329	linest_r2	367
Select statement	329	linest_seb	368
Selection Styles	142	linest_sem	367
Selections	65, 129–130	linest_sey	368
current	141	linest_ssreg	370
indicator	89	linest_ssresid	370
locking	138	median	364
moving	135	skew	365
multiple	130	stdev	364
unlocking	138	sterr	365
Semantic		steyx	366
links	483–487	Statistical distribution function in script	
statement	332	chidist	395
tables	483	chiinv	395
Serialization	36	fdist	397
Set statement	332	finv	397
Setdateyear	442	normdist	396
Setdateyearmonth	442	norminv	396
Settings menu	68	tdist	396
		tinv	397

Statistical distribution functions	
in script expressions	395
Statistical test function in script	
chi2test_chi2	373
chi2test_df	372
chi2test_p	372
ttest_df	373
ttest_dif	374
ttest_lower	374
ttest_sig	373
ttest_sterr	374
ttest_t	373
ttest_upper	375
ttest1_conf	378
ttest1_df	377
ttest1_dif	378
ttest1_lower	378
ttest1_sig	377
ttest1_sterr	378
ttest1_t	377
ttest1_upper	379
ttest1w_conf	380
ttest1w_df	379
ttest1w_dif	380
ttest1w_lower	380
ttest1w_sigTtest1w_sig	379
ttest1w_sterr	380
ttest1w_t	379
ttest1w_upper	381
ttestw_conf	376
ttestw_df	375
ttestw_dif	376
ttestw_lower	376
ttestw_sig	376
ttestw_sterr	376
ttestw_t	375
ttestw_upper	377
ztest_conf	382
ztest_dif	381
ztest_sig	381
ztest_sterr	382
ztest_z	381
ztestw_conf	383
ztestw_dif	383
ztestw_sig	382
ztestw_sterr	383
ztestw_z	382
Statistical Test Functions	372
Status bar	89
Stdev	364
Sterr	365
Steyx	366
Store	
statement	336
String aggregation functions in script	
concat	362
FirstValue	362
LastValue	362
MaxString	361
MinString	361
String functions	
in script expressions	405
String functions in script	
applycodepage	411
capitalize	407
chr	405
evaluate	410
findoneof	410
hash128	411
hash160	411
hash256	411
index	406
KeepChar	407
left	405
len	405
lower	407
ltrim	407
mid	406
ord	405
PurgeChar	408
repeat	407
replace	408
right	406
rtrim	407
subfield	408
substringcount	411
textbetween	408
trim	407
upper	406
StripComments	
variable in script	345
Sub..end sub statement	337
Subfield	408
Substringcount	411
sum	
basic aggregation function in script ...	359
Support	16
Switch..case..default..end switch statement	338

Synthetic Keys	465	Today	430
System Fields	461	Toolbar	
System fields	461	Bookmark	84
System functions		configuration	84
in script expressions	418	Design	81
System functions in script		Main	77
ComputerName	419	Sheet	84
GetRegistryString	419	Tools menu	71
Input	419	Trace	
MsgBox	419	statement	339
OSuser	418	Transform Dialog	257
qlikviewversion	418	Trigonometric and Hyperbolic functions in script	
QVuser	418	acos	394
ReloadTime	419	asin	394
		atan	394
		atan2	395
		cos	394
		cosh	395
		sin	394
		sinh	395
		tan	394
		tanh	395
		Trigonometric functions	
		in script expressions	394
		Trim	407
		Troubleshoot	38
		True	402
		Ttest_df	373
		Ttest_dif	374
		Ttest_lower	374
		Ttest_sig	373
		Ttest_sterr	374
		Ttest_t	373
		Ttest_upper	375
		Ttest1_conf	378
		Ttest1_df	377
		Ttest1_dif	378
		Ttest1_lower	378
		Ttest1_sig	377
		Ttest1_sterr	378
		Ttest1_t	377
		Ttest1_upper	379
		Ttest1w_conf	380
		Ttest1w_df	379
		Ttest1w_dif	380
		Ttest1w_lower	380
		Ttest1w_sterr	380
		Ttest1w_t	379
		Ttest1w_upper	381
T			
Table			
labels	342		
logical	462		
naming of	342		
Table functions in Script			
FieldName	425		
FieldNr	425		
NoOfFields	425		
NoOfRows	424		
NoOfTables	425		
TableName	425		
TableNumber	425		
Table Viewer	228		
TableName	425		
TableNumber	425		
Tan	394		
Tanh	395		
Tdist	396		
Text	453		
text file	315		
Text search	130		
Textbetween	408		
TextCount	363		
ThousandSep			
variable in script	347		
Time	451, 455		
Time functions			
in script expressions	426		
TimeFormat			
variable in script	348		
Timestamp	452, 456		
TimestampFormat			
variable in script	348		
Tinv	397		

Ttestw_conf	376
Ttestw_df	375
Ttestw_dif	376
Ttestw_lower	376
Ttestw_sig	376
Ttestw_sterr	376
Ttestw_t	375
Ttestw_upper	377
txt	315

U

Union operator	331
Unless statement	339
Unlocking selections	138
Unmap statement	340
Unqualify statement	341
Upper	406
User Preferences	91–124
UTC	448

V

Value Handling Variables	346
Variables in script	343–350
Verbatim variable in script	346
View menu	64

W

Week	426
Weekday	427
WeekEnd	438
WeekName	439
WeekStart	438
Weekyear	427
When statement	342
Where clause	311
While clause	311
Wild cards in data	491–492
Wildmatch	417
Wildmatch5	417
Window menu	72
WinPath variable in script	344
WinRoot variable in script	345
Working Set Limits %	95

X

Xirr	371
XML File Wizard	256
Xnpv	372
Xor	357

Y

Year	427
YearEnd	433
YearName	434
YearStart	433
Yeartodate	431

Z

Ztest_conf	382
Ztest_dif	381
Ztest_sig	381
Ztest_sterr	382
Ztest_z	381
Ztestw_conf	383
Ztestw_dif	383
Ztestw_sig	382
Ztestw_sterr	383
Ztestw_z	382



Reference Manual

Book II: Layout, Number Formats and Macros

Version 9.0 for Microsoft Windows®

Third Edition, Lund, Sweden, February 2010

Authored by QlikTech International AB /HIC/KHN/JNN/MSJ/CEN

Copyright © 1994-2010 Qlik®Tech International AB, Sweden.

Under international copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written permission of QlikTech International AB, except in the manner described in the software agreement.

Qlik®Tech and Qlik®View are registered trademarks of QlikTech International AB.

Microsoft, MS-DOS, Windows, Windows NT, Windows 2000, Windows 2003, Windows XP, Windows Vista, SQL Server, FoxPro, Excel, Access and MS Query are trademarks of Microsoft Corporation.

IBM, AS/400 and PowerPC are trademarks of International Business Machines Corporation.

Borland, Paradox and dBASE are trademarks of Borland International.

ORACLE and SQL*Net are trademarks of Oracle Corporation.

MacOS is a trademark of Apple Corporation.

Sun Java is a trademark of Sun Microsystems, Inc.





OVERVIEW BOOK II

SHEETS AND SHEET OBJECTS

I

NUMBER FORMATS

II

MACROS AND AUTOMATION

III



CONTENT

PART I: SHEETS AND SHEET OBJECTS

1	DOCUMENT PROPERTIES	15
	1.1 General	16
	1.2 Opening	26
	1.3 Sheets	29
	1.4 Server	33
	1.5 Scheduler	37
	1.6 Variables	38
	1.7 Security	42
	1.8 Triggers	45
	1.9 Groups	48
	1.10 Tables	51
	1.11 Sort	54
	1.12 Presentation	56
	1.13 Number	58
	1.14 Scrambling	59
	1.15 Font	60
	1.16 Layout	61
	1.17 Caption	68
2	THE SHEET	73
	2.1 Creation	73
	2.2 Navigation	73
	2.3 Object Menu for Sheets	74
	2.4 Sheet Properties	75
	2.5 Tabrow Properties	88
3	SHEET OBJECTS	91
	3.1 Local Objects and Server Objects	91
	3.2 Creation	91
	3.3 Navigation	92
	3.4 Object Menu for Sheet Objects	92
	3.5 Sheet Object Properties	92
	3.6 Minimized and Maximized Sheet Objects	93
	3.7 The Server Objects Dialog	93
4	THE LIST BOX	97

4.1 Creation	97
4.2 List Box Object Menu	97
4.3 List Box Properties	101
4.4 List Box Tree View	123
5 THE STATISTICS BOX	125
5.1 Creation	125
5.2 Statistics Box Object Menu	125
5.3 Statistics Box Properties	128
6 THE CHART	135
6.1 QlikView Charts	135
7 THE MULTI BOX	137
7.1 Creation	137
7.2 Multi Box Object Menu	137
7.3 Multi Box Properties	140
8 THE TABLE BOX	155
8.1 Creation	155
8.2 Using the Table Box	155
8.3 Table Box Object Menu	156
8.4 Custom Format Cell Dialog	160
8.5 Table Box Properties	162
9 THE CURRENT SELECTIONS BOX	173
9.1 Creation	173
9.2 Current Selections Box Object Menu	173
9.3 Current Selections Box Properties	176
10 THE INPUT BOX	181
10.1 Creation	181
10.2 Using the Input Box	181
10.3 Input Box Object Menu	182
10.4 Input Box Properties	184
11 THE BUTTON	195
11.1 Creation	195
11.2 Button Object Menu	195
11.3 Button Properties	196
12 THE TEXT OBJECT	211
12.1 Creation	211
12.2 Object Menu of the Text Object	211
12.3 Text Object Properties	213
13 THE LINE/ARROW OBJECT	219

13.1 Creation	219
13.2 Object Menu of the Line/Arrow Object	219
13.3 Line/Arrow Object Properties	220
14 THE SLIDER/CALENDAR OBJECT	225
14.1 Using a Slider/Calendar Object	225
14.2 Creation	226
14.3 Object Menu of the Slider/Calendar Object	226
14.4 Slider/Calendar Object Properties	228
15 THE BOOKMARK OBJECT	237
15.1 Creation	237
15.2 Object Menu of the Bookmark Object	237
15.3 Bookmark Object Properties	239
16 THE SEARCH OBJECT	243
16.1 Creation	243
16.2 Object Menu of the Search Object	243
16.3 Search Object Properties	244
17 CUSTOM OBJECT	251
17.1 Creation	251
17.2 Object Menu of the Custom Object	251
17.3 Custom Object Properties	253
18 SERVER COLLABORATION	257
18.1 The Collaboration Pane	257
18.2 The Server Objects Dialog	259
19 EXPORTING AND PRINTING	263
19.1 Print	263
19.2 Print Sheet	271
19.3 Print Preview	271
19.4 Copy Mode	272
19.5 Export to File	274
20 LAYOUT THEMES	275
20.1 About QlikView Layout Themes	275
20.2 Applying Themes in the Layout	276
20.3 Making Themes with the Theme Maker Wizard	278
21 OBJECTS CLIENT PAGE GENERATOR	287

PART II: NUMBER FORMATS

22 DATA TYPES IN QLIKVIEW	291
22.1 Data Representation inside QlikView	291
22.2 Number Interpretation	291
22.3 Dates and Times	296
23 NUMBER FORMAT DIALOGS	301
23.1 Document Properties	301
23.2 List Box, Multi Box, Table Box and Input Box	303
23.3 Chart	304
24 FORMAT CODES	307
24.1 Numbers	307
24.2 Dates	308
24.3 Times	309
24.4 Time Stamps	310

PART III: MACROS AND AUTOMATION

25 USING AUTOMATION AND MACROS WITH QLIKVIEW	313
25.1 The QlikView Automation Interface	313
25.2 How Automation and Macros Can Control QlikView	313
26 INTERNAL MACRO INTERPRETER AND MACROS	315
26.1 Edit Module Dialog	315
26.2 Macro debugger	320
26.3 Invoking Macros	322
26.4 Examples of Macros	327
26.5 Special library functions for JScript	328
27 GETTING HOLD OF A QLIKVIEW DOCUMENT	329
27.1 Accessing QlikView Documents from the Outside	329
27.2 Accessing QV Documents from the Internal Macro Interpreter	329
28 VBSCRIPT FUNCTION CALLS FROM SCRIPT	331
28.1 Transfer of Parameters	333
29 THE VISUAL BASIC PROGRAMMING LANGUAGE	335
29.1 About Visual Basic	335
29.2 Differences Between VB, VBA and VBS	335
29.3 Note About Parameters to Methods and Properties	336

29.4 Getting Hold of the System from VBScript	337
30 USING MACROS IN QV DOCUMENTS ON THE QV-SERVER	339
30.1 Macros on QlikView Server	339
30.2 Macro functionality limitations	339
30.3 Macro trigger limitations	340
30.4 VBScript functions	340
30.5 Server Side Export	340



PART I: SHEETS AND SHEET OBJECTS

- **Document Properties**
- **Sheets**
- **Sheet objects**
- **Printing and exporting**



1 DOCUMENT PROPERTIES

Document properties are settings that are stored in the document file.

The **Document Properties** dialog is opened by choosing **Document Properties** from the **Settings** menu or by choosing the **Document Properties** button from the toolbar.



When the properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

1.1 General

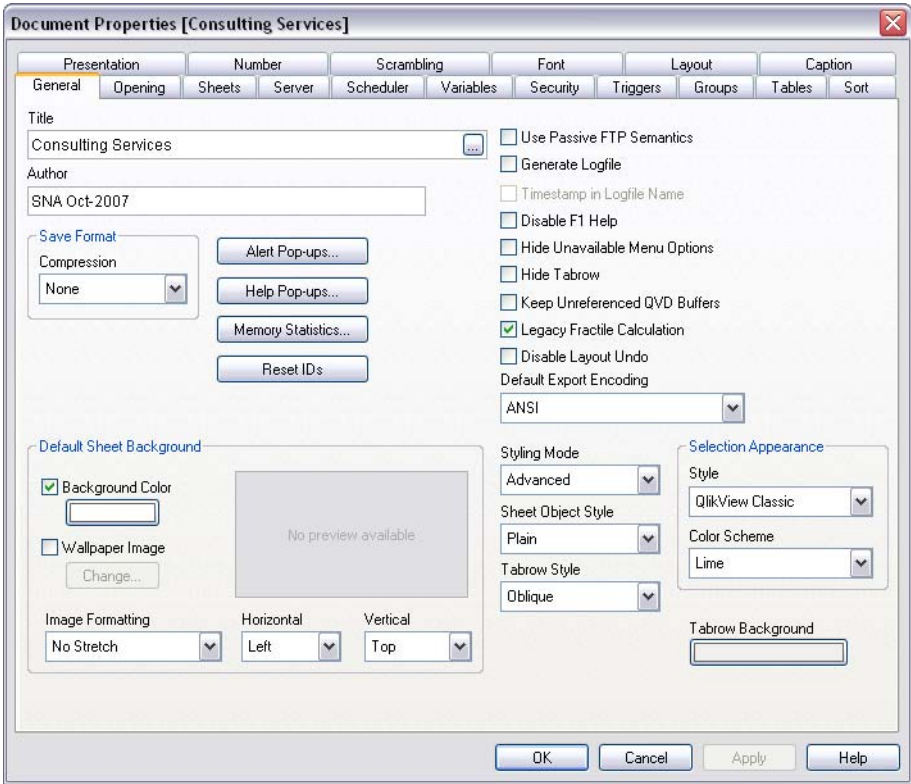


Figure 1. The General page in the Document Properties dialog

Title

By entering a text in the **Title** box, it is possible to set the window name for the document. By clicking the ... button the full **Edit Expression** dialog for easier editing of long formulas will open.

Author

Here you specify the author of the QlikView file.

Save Format

In this group you set the level of compression for the QlikView files.

Compression

This drop-down specifies the save compression mode for the document. By using compression the size of the file will be reduced by typically 60-80% (actual results will vary

with the document). When using compression, document save times will be slightly increased. With **Medium** compression all parts of the document except table data (which is already stored quite compressed inside QlikView) will undergo compression. With **High** compression (default) also table data will undergo compression, saving some space but further increasing save and load times. By choosing **None** all data are saved without compression. To set a default compression for all new files, change the **Preferred Save Format** setting in the **User Preferences** dialog, **Save** page (page 97 in Book I).

Alert Pop-ups

Click this button to open the **Pop-up Window Settings** dialog where the appearance of help pop-up messages in the document can be set. See page 21 in Book II for details on this dialog.

Help Pop-ups

Click this button to open the **Pop-up Window Settings** dialog where the appearance of help pop-up messages in the document can be set. See page 21 in Book II for details on this dialog.

Memory Statistics

This button lets you create a text file containing memory usage statistics for the current QlikView document. This file can be read e.g. by QlikView for an analysis of memory requirements for different parts of the document.

Reset IDs

This button enables you to make a complete renumbering of all sheet, sheet object, alert, report and bookmark IDs. The first sheet will be assigned SH01, the first chart CH01 etc. Before performing the operation, QlikView will prompt you for confirmation. Note that this command will break all macros which reference objects via their IDs.

Default Sheet Background

In the **Default Sheet Background** group, it is possible to set a background for the document. This is used on all sheets, which do not have a setting of their own (see page 77). A preview pane indicates current settings.

Background Color

Mark this check box and click the color button to set a background color for all the sheets in the document. The color can be defined as a solid color or a gradient via the

Color Area dialog (see page 23) that opens when clicking the button.

Wallpaper Image

Mark this check box to have a background image, a wallpaper, for the document. Click the **Change** button if you want to change the wallpaper.

Image Formatting

Sets properties for image formatting when image background is used.

No Stretch

If this option is selected, the image will be shown as is, without any stretching. This may cause parts of the picture to be invisible or only part of the sheet to be filled.

Fill

If this option is selected, the image will be stretched to fit the sheet without bothering about keeping the aspect ratio or the image.

Keep Aspect

If this option is selected, the image will be stretched as far as possible to fill the sheet while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the image will be stretched to fill the sheet in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Tile

If this option is selected, the image will be tiled as many times as space allows.

Horizontal

Sets horizontal alignment for image when image background is used.

Vertical

Sets vertical alignment when image background is used.

Use Passive FTP Semantics

Mark this check box to use passive FTP mode. With this setting, the client, not the server decides which ports to open for the file transfer

and opens them from the inside, thus avoiding potential security issues in the firewall.

Generate Log file

Generates a log file in the QlikView directory.

Timestamp in Logfile Name

Puts a timestamp in the logfile name, e.g.

sales.qvw.2009_02_26_12_09_50.log. The setting is only available if the **Generate Log File** check-box is marked.

Disable F1 Help

Mark this check box if you do not wish to invoke help when pressing F1.

Hide Unavailable Menu Options

If this check box is selected, object menu items made unavailable by security settings will be removed from the menu rather than grayed.

Hide Tabrow

If this check box is selected, the QlikView sheet tabrow will not be shown.

Keep Unreferenced QVD Buffers

Normally all automatically created QVD buffers (created via buffered prefix on **load** and **select** statements) will be removed by QlikView when no longer used by the document that created them. After a successful script execution (non partial reload) QlikView will scan all QVD buffers currently residing in the default folder for QVD buffers (see “User Preferences” on page 91 in Book I). Any buffer created by the current document and not referenced in the just finished reload will be removed. By marking this check box this purge will not occur. Note that this is not recommended and may cause unnecessary use of hard disk space.

Legacy Fractile Calculation

When marking this check box, QlikView will use discrete values as results from the fractile aggregation function. When this option is deselected QlikView will use an interpolated value, as known from the Percentile function in Microsoft Excel. QlikView versions before 7.5 only had support for the discrete value algorithm.

Disable Layout Undo

When this check box is marked the layout undo buffer will be suspended. This may be useful to prevent unnecessary memory consumption when deploying certain QlikView documents. Each time a user or a macro changes the layout a chunk of data is normally added to the layout undo buffer. In extreme cases, the accumulated

buffer data may cause problems when deploying documents on a QlikView Server.

Default Export Encoding

Sets the default character set for export from the document. You may choose between ANSI, Unicode and UTF-8.

Styling Mode

In this drop-down menu you can choose the object style for all your sheet objects.

Advanced mode makes it possible to configure most settings, such as the **Sheet Object Style** and the type of border to be used. These individual settings are configured on the **Layout** page of the objects.

Simplified mode also gives you the choice of **Sheet Object Style**, but configures most settings automatically. It sets such things as scrollbars and borders. Some settings can still be changed on the **Layout** page of the objects.

Sheet Object Style

Select one of the available styles for sheet object captions in this dropdown. The style selected will be used for all sheet objects with captions in the document.

Tabrow Style

Select one of the available styles for the tabrow appearance in this drop-down. The style selected will be used for all tabs in the document.

Tabrow Background

Click this button to select a custom color for the area behind the sheet tabs.

Selection Appearance

QlikView supports several different ways of presenting data and making selections in list boxes and multi boxes. The **QlikView Classic, Corner Tag, LED and Tube** styles all use color coding for indication of selected, possible and excluded values. The **Windows Check Boxes** style mimics the standard Windows interface with check boxes at each value. By choosing a specific style in this control, you may force the document to appear in one style or another wherever opened. More information about the selection styles can be found on page 142 in Book I.

When using the selection styles based on color, there are a number of different color schemes available. The basic colors (green for

selected, blue for locked etc.) cannot be changed, but variations of tone and intensity are possible.

Style

Sets the selection style for the document. Select between the available alternatives in the drop-down list. By choosing **<user default>** the document will always be opened with the selection style set as preferred by the user under **User Preferences** (see page 91 in Book I) on the computer where it is opened.

Color Scheme

Sets the selection color scheme for the document. Select between the available alternatives in the drop-down list. By choosing **<user default>** the document will always be opened with the selection color scheme set as preferred by the user under **User Preferences** (see page 91 in Book I) on the computer where it is opened.

Pop-up Window Settings

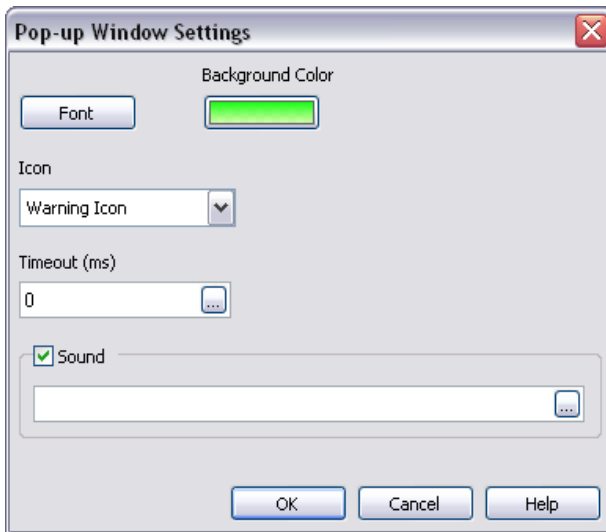


Figure 2. The Pop-up Window Settings dialog

This dialog is used to customize the pop-up messages appearing in various parts of the layout.

Font

Opens the **Font** dialog where you may set the font to be used for the pop-up message.

Background Color

Opens the **Color Area** dialog where you may set the background to be used for the pop-up message.

Icon

In this drop-down you can select an icon to be shown in the pop-up.

Time-outtime-out (ms)

Here you may enter a time in milliseconds after which the pop-up will automatically disappear from the screen. By setting the time-out to 0 the pop-up will stay on the screen until the user clicks it away.

Sound

Mark the check box to use a sound with the pop-up. The edit box below should contain a valid path to a .wav sound file. This may be an external path (e.g. c:\mysound.wav) or a path to a bundled QlikView sound (e.g. qmem://<bundled>/sounds/qv_ok.wav). For a list of bundled sounds see “Bundled Sounds” on page 208 in Book I).

Color Area

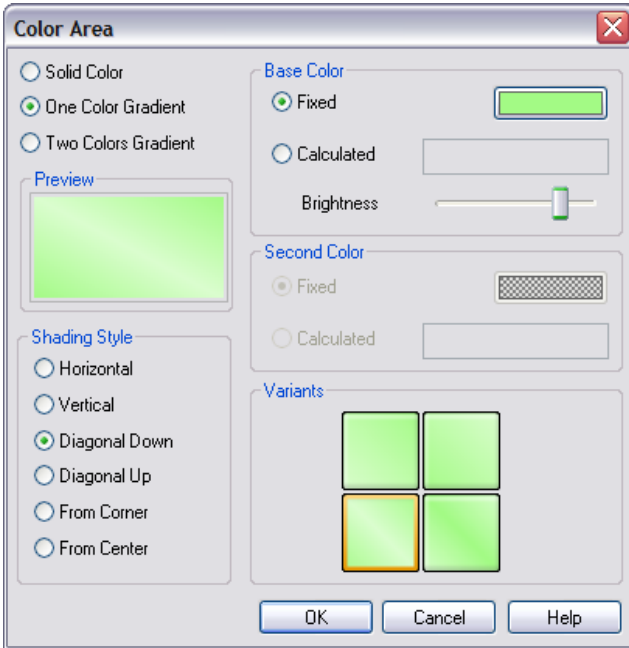


Figure 3. The Color Area dialog

Most colored surfaces in a QlikView layout can be formatted either with a solid color covering the entire area or with a gradient effect. Color(s) can be fixed or defined by a calculated expression dynamic update. All these settings are made in the **Color Area** dialog described below. This dialog is referenced from numerous places throughout this book.

Preview

As you make selections in the dialog this pane offers you a preview of how the colored surface will look in the layout.

Solid Color

Select this basic option to have one single color covering the entire surface to be formatted. The color can be fixed or calculated as specified in the **Base Color** group described below.

One Color Gradient

Select this basic option to have surface formatted with a gradient based on varying brightness of one color. The color can be fixed or calculated as specified in the **Base Color** group described below.

Two Color Gradient

Select this basic option to have surface formatted with a gradient between two selected colors. The colors can be fixed or calculated as specified in the **Base Color** and **Second Color** groups described below.

Base Color

This group is used to determine the color for solid surfaces and the base color for gradients.

Fixed

If you want a fixed color, select this radio button. Click the colored button to select color.

Calculated

If you want a calculated color, select this radio button. Enter an expression formula in the edit box. The expression must evaluate to a valid color representation (a number representing the Red, Green and Blue components), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used. The color may contain an alpha factor for transparency.

Brightness

This control is only available for one-color gradients. It sets the relative brightness of the opposite end of the gradient. The center position will render the surface as if a solid color was selected. Set to the left of the center the gradient will go towards a darker shading of the base color. To the right the gradient will go towards a lighter shading.

Second Color

This group is used to determine the second color for two-color gradients.

Fixed

If you want a fixed second color select this radio button. Click the colored button to select color.

Calculated

If you want a calculated second color select this radio button. Enter an expression formula in the edit box. The expression must evaluate to a valid color representation (a number representing the Red, Green and Blue components), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the

expression is not a valid color representation, black will be used. The color may contain an alpha factor for transparency.

Shading Style

In this group you select the direction of the gradient when one of the gradient options is used.

Horizontal

Select this radio button for a horizontally oriented shading pattern.

Vertical

Select this radio button for a vertically oriented shading pattern.

Diagonal Down

Select this radio button for a diagonally oriented shading pattern reaching from the upper left to the lower right.

Diagonal Up

Select this radio button for a diagonally oriented shading pattern reaching from the lower left to the upper right.

From Corner

Select this radio button for a shading pattern reaching from a selected corner.

From Center

Select this radio button for a shading pattern reaching from the center.

Variants

This group allows you to choose between a number of variants based on the **Shading Style** selected above.

1.2 Opening

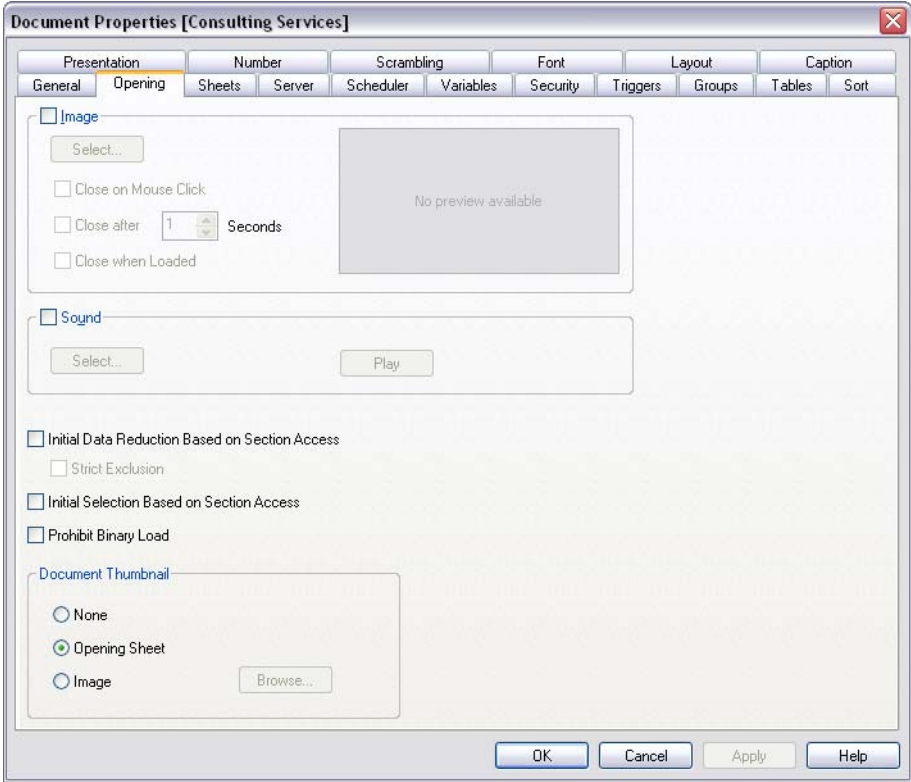


Figure 4. The Opening page in the Document Properties dialog

In this page it is possible to customize the opening of a document.

Image

Mark this check box to have an opening image for the document.

Select...

Click this button to open the **Select Opening Image** dialog where you can select an image (bmp, jpg, jpeg, or png).

Close on Mouse Click

Lets the user close the opening picture by clicking it. Either this check box or the one below must be marked.

Close after N Seconds

Closes the opening picture automatically. Either this check box or the one above must be marked.

Close when Loaded

Closes the opening picture when the file has been loaded.

Sound

Mark this check box to have an opening sound for the document. The **Select Opening Sound** dialog will open.

Initial Data Reduction Based on Section Access

If this option is selected, QlikView will perform a procedure on opening the document (or on establishing a new session on QlikView Server) which permanently hides certain data from the user based on the identity with which the user has logged on to the document. See page 505 in Book I for details. When this feature is used on a document not run exclusively on a QlikView Server it should always be combined with the **Prohibit Binary Load** option (see below) in order to maintain data access security. This setting is mutually exclusive with **Initial Selection Based on Section Access**.

Strict Exclusion

If this check box is marked in combination with **Initial Data Reduction Based on Section Access**, strict exclusion will be used when reducing the data. This means that access to the document will be denied whenever the field values in the section access reduction fields lack matches in their corresponding section application field. This, however, does not apply for users with Admin status, who instead will see the unreduced data set if there are no matches. This setting is turned on by default in documents created with QlikView 7.02 and later and is recommended for maximum access security.

Initial Selection Based on Section Access

If this option is marked, QlikView will perform a procedure on opening the document (or on establishing a new session on QlikView Server) which shows the selection based on the identity with which the user has logged on to the document. See page 505 in Book I for details. When this feature is used on a document not run exclusively on a QlikView Server it should always be combined with the **Prohibit Binary Load** option (see below) in order to maintain data access security. This setting is mutually exclusive with **Initial Data Reduction Based on Section Access**.

Prohibit Binary Load

If this option is selected it will not be possible to load data from the document's qvw file via a binary statement in another QlikView document.

Document Thumbnail

Sets how the document will be displayed on the AccessPoint when the option **Thumbnail** is chosen.

None

A thumbnail with the message **Missing Image** will be displayed on the AccessPoint.

Opening Sheet

The opening sheet of the document will displayed on the AccessPoint.

Image

Click on the **Browse** button to find the image you want to display on the AccessPoint.

1.3 Sheets

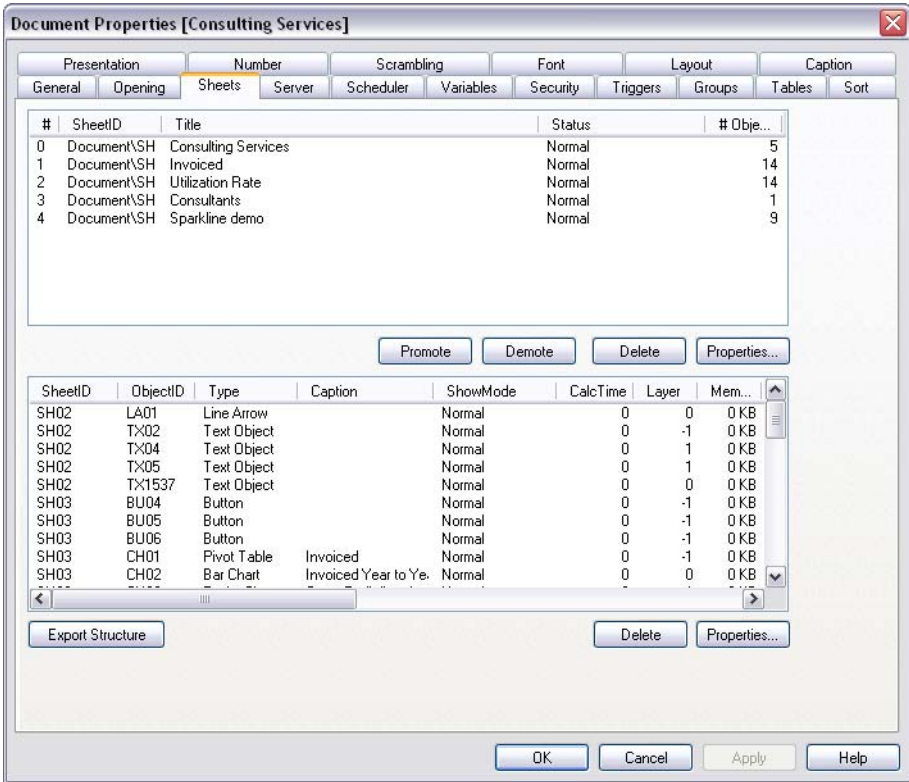


Figure 5. The Sheets page in the Document Properties dialog

This page shows two lists. The **Sheets** list at the top shows all sheets of the QlikView document. The columns are as follows:

#

The number (position) of the sheet starting with 0.

SheetID

The unique ID of the sheet.

Title

The text currently shown in the sheet tab.

Status

The current status of the sheet. Indicates if the sheet has a show condition and if it is visible (Normal) or hidden.

Objects

The current number of sheet objects on the sheet.

The list can be sorted on any column by clicking on the column header. You may also select a specific sheet by clicking on it in the list. The following options then become available as buttons below the list:

Promote

Promotes the selected sheet one step to the left (up the list). The sheet immediately moves in the list but the actual promote will not take place until you select **OK** or **Apply**.

Demote

Demotes the selected sheet one step to the right (down the list). The sheet immediately moves in the list but the actual demote will not take place until you select **OK** or **Apply**.

Delete

Deletes the selected sheet from the document. The sheet immediately disappears from the list but the actual delete will not take place until you select **OK** or **Apply**.

Properties

Opens the **Properties** dialog for the selected sheet.

The **Sheet Objects** list at the bottom of the page shows all sheet objects of the QlikView document or, if one sheet has been selected in the list above, the sheet objects of that sheet. The columns are as follows:

SheetID

The unique ID of the sheet containing the sheet object.

Object ID

The unique object ID of the sheet object.

Type

The type of the sheet object.

Caption

The text currently shown in the caption of the sheet object.

ShowMode

The current status of the sheet object. Indicates if the object is minimized, maximized, hidden or has a show condition.

CalcTime

The amount of time in milliseconds needed for the last recalculation of the object's contents. This information is only relevant for charts, tables and objects containing one or more formulas.

Layer

The current layout layer of the sheet object. The layers are set on the **Layout** page of sheet object properties dialog (**Bottom, Normal, Top** and **Custom**), where **Bottom, Normal** and **Top** correspond to the numbers -1, 0 and 1. Choose **Custom** to enter a value of your choice. Valid layer numbers are -128 to 127.

Memory

The amount of memory in kBytes needed for the last recalculation of the object's contents. This information is only relevant for charts, tables and objects containing one or more formulas.

Left

The current horizontal position of the left edge of the sheet object, expressed in pixels.

Top

The current vertical position of the top edge of the sheet object, expressed in pixels.

Right

The current horizontal position of the right edge of the sheet object, expressed in pixels.

Bottom

The current vertical position of the bottom edge of the sheet object, expressed in pixels.

Width

The current width of the sheet object, expressed in pixels.

Height

The current height of the sheet object, expressed in pixels.

Left (Min)

The current horizontal position of the left edge of the sheet object's minimized icon, expressed in pixels.

Top (Min)

The current vertical position of the top edge of the sheet object's minimized icon, expressed in pixels.

Right (Min)

The current horizontal position of the right edge of the sheet object's minimized icon, expressed in pixels.

Bottom (Min)

The current vertical position of the bottom edge of the sheet object's minimized icon, expressed in pixels.

Width (Min)

The current width of the sheet object's minimized icon, expressed in pixels.

Height (Min)

The current height of the sheet object's minimized icon, expressed in pixels.

Export Structure

By pressing this button you can export the sheet and sheet object structure of the document to a set of text files. These text files, one for the sheets (*filename.Sheets.tab*), and one for the sheet objects (*filename.Objects.tab*) can easily be read back into QlikView for further analysis with the full power of the QlikView logic. A dialog will appear where you can choose the target folder for the export. The default is to put the files in the same folder as the QlikView document.

The list can be sorted on any column by clicking on the column header. You may also select a specific sheet object by clicking on it in the list. The following two options then become available as buttons below the list:

Delete

Deletes the selected sheet object from the sheet. The object immediately disappears from the list but the actual delete will not take place until you select **OK** or **Apply**.

Properties

Opens the **Properties** dialog for the selected sheet object.

1.4 Server

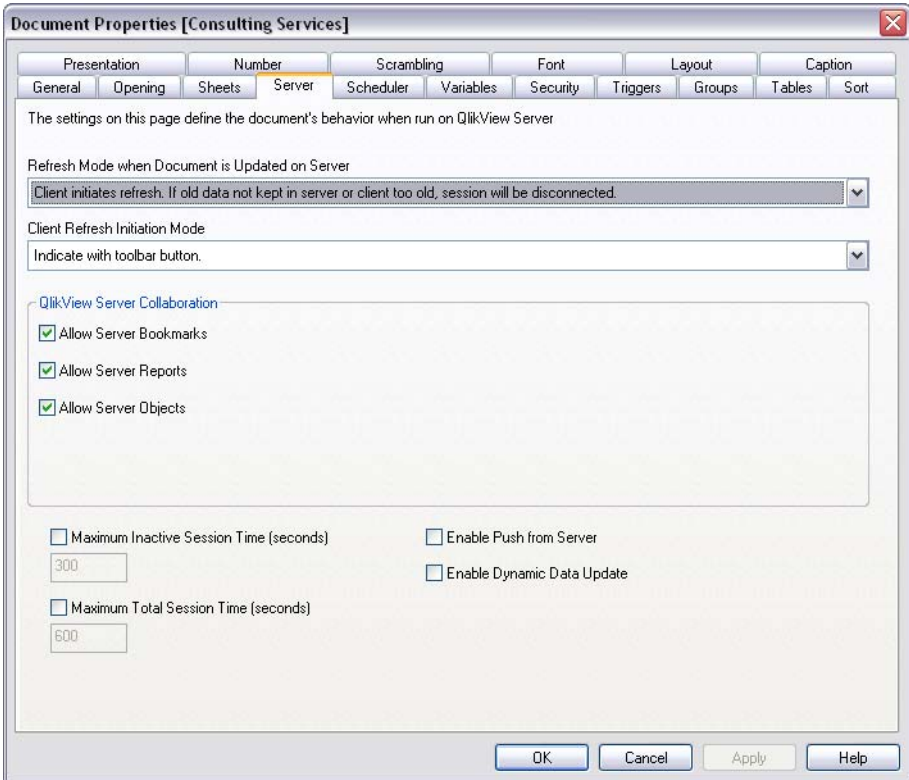


Figure 6. The Server page of the Document Properties dialog

On this page you define certain aspects of the document's behavior when run on a QlikView Server.

Refresh Mode when Document is Updated on Server

This setting defines how an updated document on the QlikView Server will be handled.

Client initiates refresh. If old data not kept in server or client too old, session will be disconnected

This is the option to choose if you never want the server to initiate a refresh of data in the client automatically. When a new version of the document becomes available on the server, the client will be offered to refresh according to the preferred set under **Client Refresh Initiation Mode** below. If this is not possible, either because the client is too

old (pre version 8) or if the server is set with **Allow only one copy of document in memory** enabled in the QlikView server control panel, the session will simply be terminated.

Client initiates refresh. If old data not kept in server or client too old, refresh will be performed automatically

This is the option to choose if you prefer to let the client initiate a refresh of data in the client when possible but will allow server to initiate a refresh automatically if necessary. When a new version of the document becomes available on the server, the client will be offered to refresh according to the preferences set under **Client Refresh Initiation Mode** below. If this is not possible, either because the client is too old (pre version 8) or if the server is set with **Allow only one copy of document in memory** enabled in the QlikView server control panel, the refresh will be performed automatically by the server.

Server performs refresh automatically, without client action

This is the option to choose if you prefer to always let the server initiate a refresh of data in the client automatically. When a new version of the document becomes available on the server, the data in the client will be refreshed.

Client Refresh Initiation Mode

This setting defines how the client should be notified of the availability of new data on the server and initiate a refresh. This setting is only valid when client initiated refresh has been selected in **Refresh Mode when Document is Updated on Server** above.

Indicate with toolbar button

The availability of new data is indicated by the **Refresh** button in the toolbar turning green and being enabled (and the corresponding **File** menu command being enabled). The refresh is performed at a time of the user's choice by clicking the toolbar button or by means of the **File** menu **Refresh** command.

Prompt with dialog

In addition to the **Refresh** button in the toolbar turning green the availability of new data will be indicated by a prompt dialog. In the dialog the user may choose between immediate refresh or wait, in which case the refresh can be performed at any later time of the user's choice by clicking

the toolbar button or by means of the **File** menu **Refresh** command.

Just do it, but tell the user it's happening

The refresh will be initiated as soon as the server indicates that new data are available. A dialog will be shown to indicate that the refresh is taking place.

Just do it

The refresh will be initiated as soon as the server indicates that new data are available. No dialog will be shown to indicate that the refresh is taking place. The user will experience a document “freeze” lasting from fractions of a second up to minutes depending on server settings and size of document.

QlikView Server Collaboration

The settings in this group make it possible to disable the clients' ability to create and share certain entities when running the document on QlikView Server. Please note that it is possible to disable all types of collaboration objects for all documents on a server by disabling the **Allow Server Collaboration** setting in the QlikView Server control panel. That setting overrides the settings below.

Allow Server Bookmarks

This check box must be enabled, if remote clients are to be allowed to create and share bookmarks with this document on the QlikView Server.

Allow Server Objects

This check box must be enabled, if remote clients are to be allowed to create and share sheet objects with this document on the QlikView Server.

Allow Server Reports

This check box must be enabled, if remote clients are to be allowed to create and share reports with this document on the QlikView Server.

QlikView Server Time-outs

If a document is published on a QlikView Server, it could be a good idea to set a maximum session time for security purposes and have the server terminate sessions that are inactive or too long. The settings at the bottom of this page allow you to do this on a document level. Hence, it is possible to have harder restrictions for documents with sensitive data than for other documents.

Corresponding settings are also found on the server side, valid for all published documents. The more restrictive of the two settings will be used.

Maximum Inactive Session Time (seconds)

This check box enables an inactive session time-out, i.e. that the server will terminate a session where the user has not clicked in a given time. The length of the inactive session time-out is set in seconds.

Maximum Total Session Time (seconds)

This check box enables a total session time-out, i.e. that the server will terminate a session after a given time, independently of whether the user has clicked or not. The length of the total session time-out is set in seconds.

Enable Push from Server

Mark this check box if the Server should allow graceful document refresh.

Enable Dynamic Data Update

Mark this check box if the Server should allow dynamic updates in a document.

1.5 Scheduler

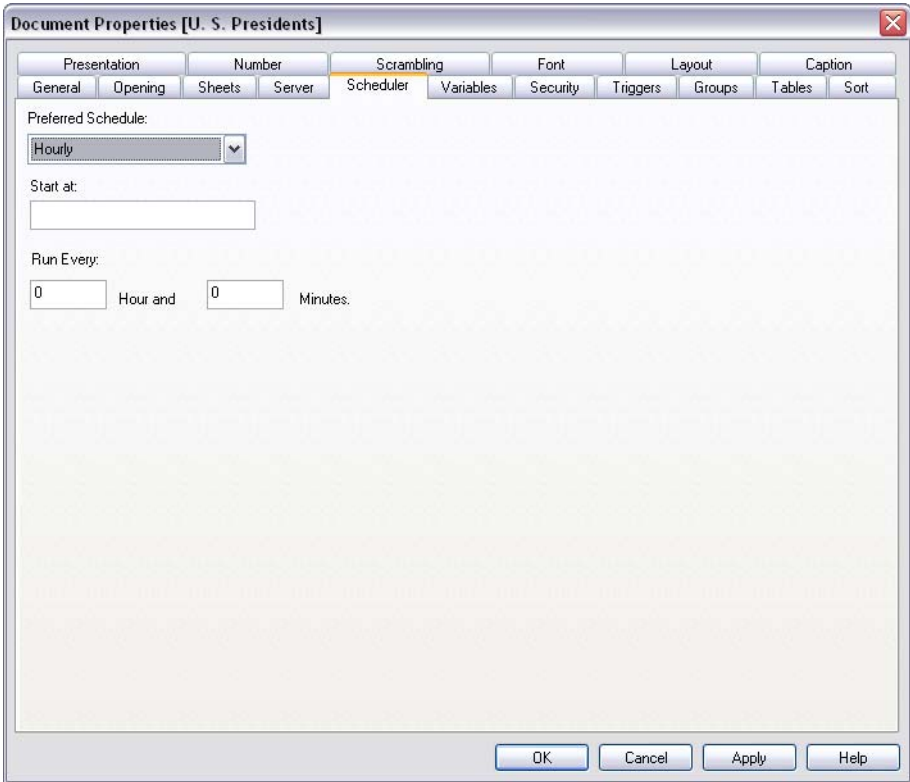


Figure 7. The Scheduler page in the Document Properties dialog

On this page a default reload schedule for the document in QlikView Server can be proposed.

Preferred Schedule

In this drop-down menu you may set the schedule for how often the document should be reloaded. The schedule can be set to **None**, **Run Once**, **Hourly**, **Daily**, **Weekly** and **Monthly**. You may then set the details of the schedule, such as start time, days of the week etc, in the fields below.

1.6 Variables

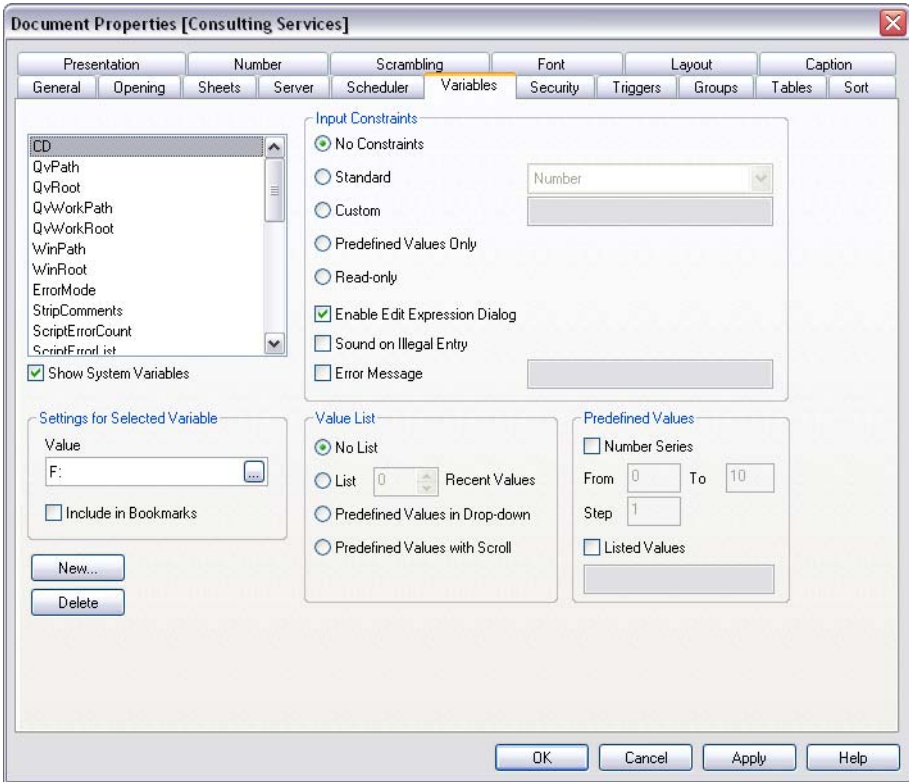


Figure 8. The Variables page in the Document Properties dialog

In this page settings can be made for the QlikView variables. For more information on variables, see page 343 in Book I.

Variables

Shows a list of all the variables in the current document. When a variable name in the list is selected the properties of that variable can be modified.

Show System Variables

If this check box is selected the system variables created by the script will be shown in the list.

Settings for Selected Variable

In this group you can make settings for a variable selected in the list above.

Value

Shows the current value of the selected variable, which can also be edited.

Include in Bookmarks

Variables are normally not included in bookmarks. Select this check box if you want the variable to be included in bookmarks.

New

Opens the **New Variable** dialog, where a new variable can be defined.

Delete

Deletes the selected variable.

Input constraints

In the **Input constraints** group you can specify constraints against which all values entered by the user into an input box variable will be checked. If a value does not meet the constraints specified, it will be rejected and an error message may be shown.

No Constraints

As default no constraints are selected, i.e. all values can be entered into the variable.

Standard

Input values will be checked against one of a number of common standard constraints, which can be selected in the drop-down box. The standard constraints are: **Integer**, **Number**, **Positive Integer**, **Positive Number** and **Single Character**. As default no constraint at all is selected, i.e. any value can be entered into the variable.

Custom

Input values will be checked against a constraint formula specified by the user. The constraint is entered in the edit box and should be expressed as a QlikView expression returning true (a non-zero value) if the input value is acceptable. The input value is referenced as a dollar sign, \$, in the expression (Example: $\$ > 0$ will cause the variable to accept only positive numbers). The previous value of the variable can be referenced by the variable name (Example: $\$ \geq \text{abc} + 1$ as a constraint on a variable named abc will accept only numeric entries with a value of the old value plus 1).

Predefined Values Only

Input values will be checked against a list of values defined in the **Predefined Values** group (see below). Only input values found in the list will be accepted.

Read-only

Marks the variable as read only. No values can be entered.

Enable Edit Expression Dialog

Mark this check box to enable expansion of the input box edit window to the full **Edit Expression** dialog by means of a ... button.

Sound on Illegal Entry

If this check box is selected, QlikView will issue a sound warning when the user attempts to enter a value which falls outside of the constraints.

Error Message

Normally, if the user attempts to enter a value which falls outside of the constraints, it will just be rejected leaving the present variable value in place. By selecting this check box it is possible to specify an error message, which will be presented to the user in case of incorrect entries. The error message is typed in the corresponding edit box. It may be defined as a calculated formula (see page 467) for dynamic update.

Value List

In the **Value List** group you can specify constraints against which all values entered into a variable will be checked. If a value does not meet the constraints specified and **Input Constraints** is set to **Predefined Values Only**, it will be rejected and an error message may be displayed.

No List

No list of values previously used in the variable is maintained. No drop-down with previous values or predefined values will be available to the user for the selected variable in input boxes.

List N Recent Values

A list of values previously used in the variable is maintained. A drop-down list with the most recently used values will be available to the user for the selected variable in the input box. The number of previous values to store can be set in the edit box.

Predefined Values in Drop-down

A drop-down list with predefined values (see **Predefined Values** below) will be available to the user for the selected variable in input boxes.

Predefined Values with Scroll

A scroll control will be available to the user for the selected variable in input boxes. The scrolling will be made between the predefined values (see **Predefined Values** below).

Predefined Values

In the **Predefined Values** group you can specify a list of predefined values which can be presented to the user in a drop-down list in input boxes and/or used to define acceptable variable values.

Number Series

If this check box is selected a list of predefined numeric values will be generated based on a lower limit, an upper limit and a step value. This option can be used alone or in combination with **Listed values** (see below).

From

Defines the lower limit for the number series. Must be a real number.

To

Defines the higher limit for the number series. Must be a real number.

Step

Defines the step between values in the number series. Must be a real number.

Listed Values

By selecting this check box it is possible to specify a list of arbitrary predefined values. The values can be numeric or alphanumeric. Alphanumeric values should be enclosed in quotes (e.g. 'abc'). Each value should be separated by a semicolon ; (e.g. 'abc';45;14.3;'xyz'). This option can be used alone or in combination with **Number series** (see above).

1.7 Security

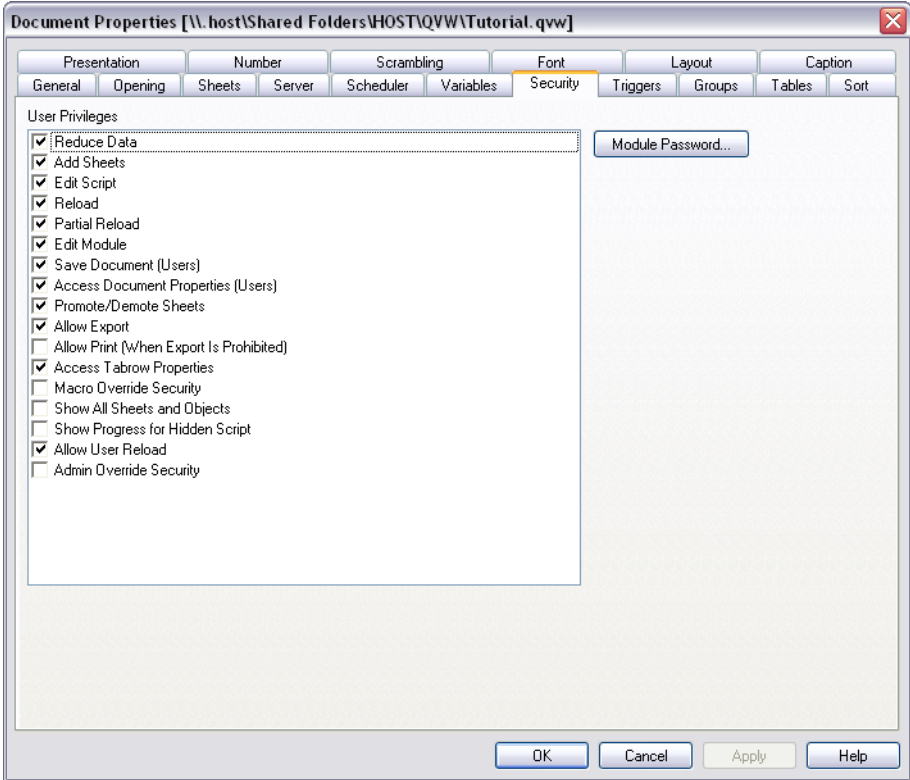


Figure 9. The Security page in the Document Properties dialog

This page is useful for documents with access restriction (for more information see page 505 and forward in Book I). By deselecting an option in the list, the ADMIN user inactivates that particular command, thus preventing undesired changes. The **Security** page is not available for people with USER privileges. By default, all the options except **Macro Override Security**, **Show All Sheets and Objects** and **Admin Override Security** are selected, i.e. active.

Reduce Data

If this check box is left unchecked, the **Reduce Data** command in the **File** menu (see page 61 in Book I) becomes inactive.

Add Sheets

If this check box is left unchecked, the **Add Sheet** command in the **Layout** menu (see page 66 in Book I) becomes inactive.

Edit Script

If this check box is deselected, the **Edit Script** command in the **File** menu (see page 61 in Book I) and in the toolbar becomes inactive.

Reload

If this check box is deselected, the **Reload** command in the **File** menu (see page 61 in Book I) and in the toolbar becomes inactive.

Partial Reload

If this check box is deselected, the **Partial Reload** command in the **File** menu (see page 61 in Book I) becomes inactive.

Edit Module

If this check box is deselected, the **Edit Module** command in the **File** menu (see page 71 in Book I) becomes inactive.

Save Document (Users)

If this check box is deselected, the **Save** command in the **File** menu (see page 59 in Book I) becomes inactive for persons with USER privileges.

Access Document Properties (Users)

If this check box is deselected, the **Document Properties** command in the **Settings** menu (see page 68 in Book I) becomes inactive for persons with USER privileges.

Promote/Demote Sheets

If this check box is deselected the **Promote Sheet** and **Demote sheet** commands in the **Layout** menu (see page 66 in Book I) become inactive.

Allow Export

If this check box is deselected all **Export**, **Print** and **Copy to Clipboard** commands become unavailable.

Allow Print (When Export Is Prohibited)

Although the check box **Allow Export** above is deselected all **Print** commands remain available if this check box is selected.

Access Tabrow Properties

If this check box is deselected the **Tabrow Properties** (see page 88) can no longer be accessed.

Macro Override Security

If this check box is selected, it is possible to override all security settings by means of macros and commands via the Automation API.

Show All Sheets and Objects

If this check box is selected all conditional display of sheets and sheet objects is overruled, so that all sheets and sheet objects

become visible. This functionality can be toggled by pressing CTRL+SHIFT+S.

Show Progress for Hidden Script

If this check box is selected the script progress dialog will be shown also while executing hidden script.

Allow User Reload

If this check box is deselected it will not be possible to reload the script when the document is opened in USER mode, even if the **Reload** check box above is selected..

Admin Override Security

If this check box is selected all security settings for the document and sheets will be disregarded while in ADMIN mode.

Module Password

This button allows you to set a password for entering the **Edit Module** dialog (see page 315). To change the password, click the **Module Password** button again.

1.8 Triggers

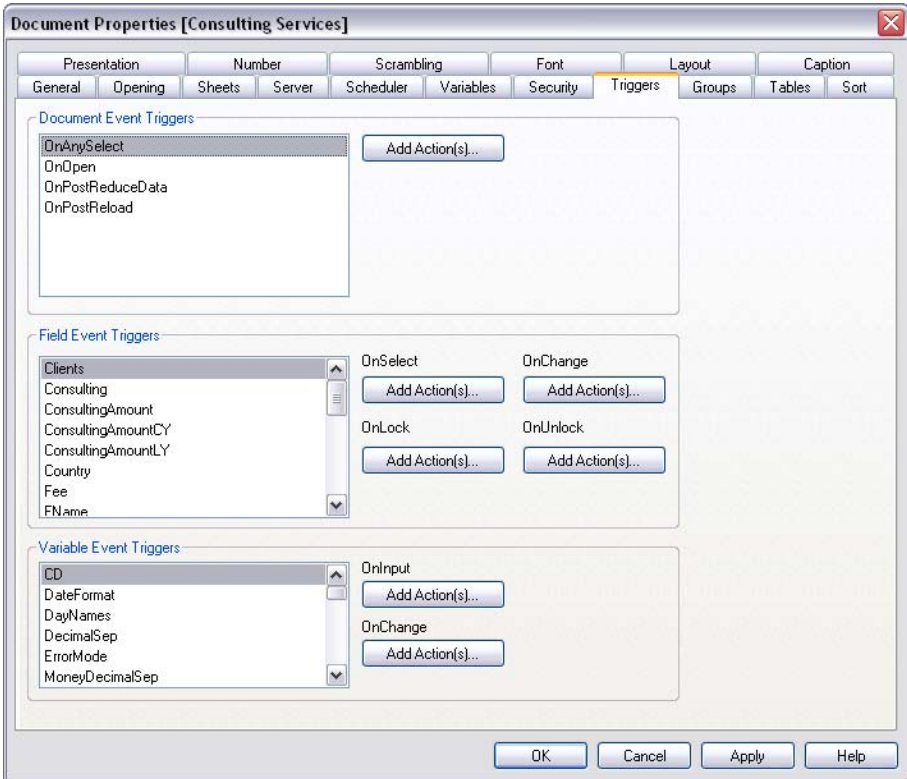


Figure 10. The Triggers page in the Document Properties dialog

On the **Triggers** page, you can set actions (including macro actions) to be invoked on document events, variable events and field events. For more information on macros, see “macros and automation” on page 311.

Not all actions can be invoked from the following event triggers. Read more about the different actions on page 200.

Note There are limitations as to the behavior of certain macro triggers when working with documents on QlikView Server. Please study the special section "Considerations when using macros in QV documents on the QlikView Server" in the QlikView Server reference manual before designing server documents with macro triggers.

Note Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

Document Event Triggers

In the **Document Event Triggers** group you can set actions to trigger on selected events in the document. You must select an event in the list before you can assign actions or macros to it.

OnAnySelect

Select this event to assign an action to be executed each time a selection has been made in any field of the QlikView document.

OnOpen

Select this event to assign an action to be executed each time the QlikView document is opened.

OnPostReduceData

Select this event to assign an action to be executed after each time the **Reduce Data** command has been executed.

OnPostReload

Select this event to assign an action to be executed each time the script has been re-executed.

Add Action(s)

This button opens the **Actions** page. On this page you can add an action to the trigger. Choose the action **macro** to assign an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. See “Actions” on page 200 for how to add actions. The action will be executed each time the selected document event occurs. When an action has been assigned to the event, the button changes to **Edit Action** and you can change the action for the event.

Field Event Triggers

In the **Field Event Triggers** group you can set actions to trigger on changes in the logical state of a specified field in the document. Select a field in the list and press one of the buttons to assign an action to trigger on events in it. The buttons open the **Actions** page. Here you can assign an action, including an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. See “Actions” on page 200 for how to add actions.

When an action has been assigned to the field, the button changes to **Edit Action**. The following **Field Event Triggers** exist:

OnSelect

The action will be executed each time a selection has been made in the specified field.

OnLock

The action will be executed each time the field is locked.

OnChange

The action will be executed each time a selection has been made in any field which is logically associated with the specified field.

OnUnlock

The action will be executed each time the field is unlocked.

Variable Event Triggers

In the **Variable Event Triggers** group you can set actions to trigger on changes in the contents of a specified variable in the document. Select a variable in the list and press one of the buttons to assign an action to it. The buttons open the actions page. On this page you can add an action to the trigger. Choose the action **macro** to assign an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. See “Actions” on page 200 for how to add actions. When an action has been assigned to the variable, the button changes to **Edit Action** and you can change the action for the event. The following **Variable Event Triggers** exist:

OnInput

The action will be executed each time a new value is directly entered in the selected variable.

OnChange

The action will be executed each time the value of the selected variable changes as a result of changes in other variables or the logical state of the document. This typically applies when the variable contains a formula.

Note For backward compatibility to earlier versions of QlikView, the action must consist of a macro action. An action that consists of only one macro action is translated back to old string format on save.

1.9 Groups

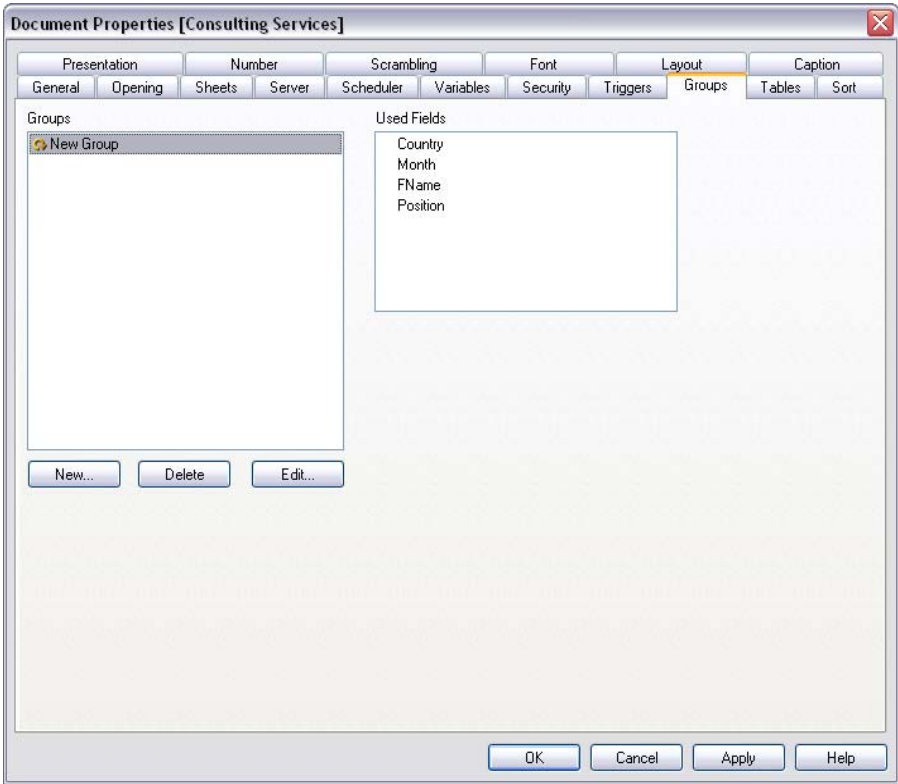


Figure 11. The Groups page in the Document Properties dialog

In this page it is possible to create drill-down or cyclic field groups. For more information on how to use groups, see page 469.

A list shows all groups in the document. An icon to the left of each group name indicates whether it is a drill-down group or a cyclic group. A group can be selected by clicking in the list. The selected groups are shown in the **Used Fields** area.

New

Clicking this button opens the **Group Settings** dialog where you define a new field group. Select the fields to include in the group from the list appearing under **Available Fields**.

Delete

Deletes the selected group.

Edit

Opens the **Group Settings** dialog for the selected group.

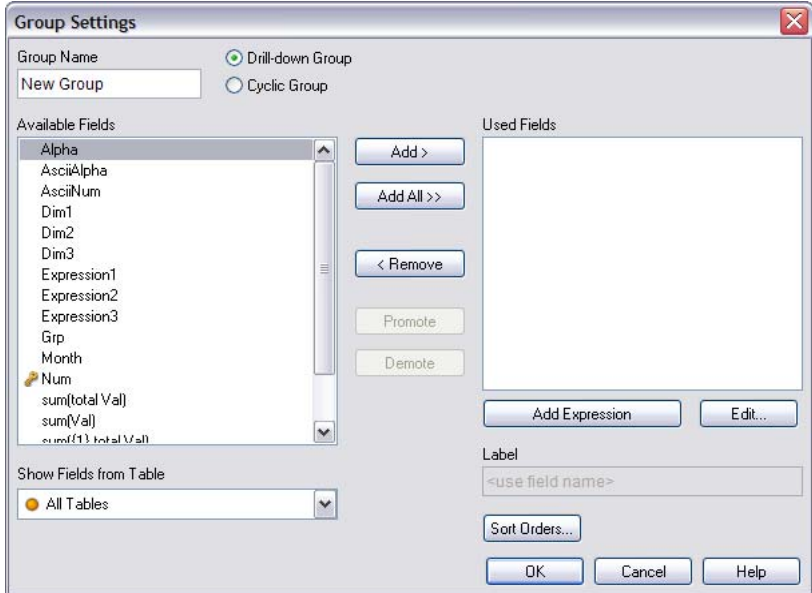
Group Settings dialog

Figure 12. The Group Settings dialog

Group Name

Change the name by entering the name you would like to attribute to your field group.

Drill-down Group

Select this option to create the group as a drill-down group.

Cyclic Group

Select this option to create the group as a cyclic group.

Available Fields

Available fields are listed in this column. Select the fields you wish to include in your group.

Used Fields

Fields included in the current group are listed in this column. Initially this column is empty.

Add >

Moves the selected fields from the column of available fields to the column of used fields.

Add All >>

Moves all the fields in the column of available fields to the column of used fields.

< Remove

Moves the selected fields listed in the column of used fields to the column of available fields.

Select the fields to be used/removed by clicking them. Use the **Add >** or the **< Remove** button to move them to the desired column.

Promote

Moves the selected field one step further up in the list of used fields.

Demote

Moves the selected field one step further down in the list of used fields.

Add Expression

Opens the **Edit Expression** dialog for easy editing of long formulas.

Edit

Opens the **Edit Expression** dialog for editing the selected formula in the **Used Fields** area.

Label

An alternative name for the displayed field can be entered here. It will be used as dimension name in the chart. The label may be defined as a calculated formula (see page 467) for dynamic update.

Sort Orders...

Clicking on this button opens the **Group Sort Orders** dialog in which a separate sort order can be set for each field in the group.

1.10 Tables

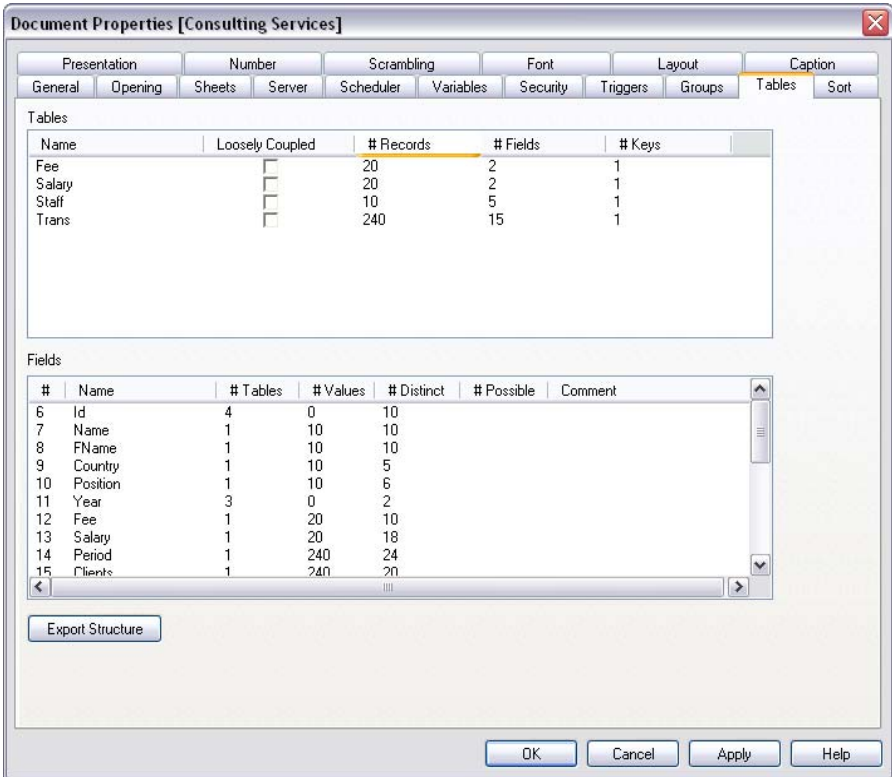


Figure 13. The Tables page in the Document Properties dialog

This page shows two lists. The **Tables** list at the top shows all input tables of the QlikView document. The columns are as follows:

Name

The name of the input table

Loosely Coupled

One or more of the tables can be made loosely coupled by marking the check box to the right of the table name. A loosely coupled table is one where the normal QlikView internal logic has been disconnected internally. This means that selections in one field do not propagate through to the other fields in the table. This can e.g. be useful for avoiding circular references in the data structure. For

more examples of how this feature can be used, please see “Loosely Coupled Tables” on page 144 in Book I.

Records

The number of records (rows) in the input table.

Fields

The number of fields (columns) in the input table.

Keys

The number of fields (columns) in the input table which are keys, i.e. appear also in other input tables.

The list can be sorted on any column by clicking on the column header. You may also select a specific sheet by clicking on it in the list.

The **Fields** list at the bottom of the page shows all fields of the QlikView document or, if one input table has been selected in the list above, the fields of that table. The columns are as follows:

#

The internal number of the fields. Numbers 0 to 5 are used for the QlikView system fields, which are not shown in the list.

Name

The name of the field.

Tables

The number of tables in which the field appears.

Values

The total number of values in the field, disregarding selections. This value is not defined for certain key fields (see “Frequency Information in Associating Fields” on page 464 in Book I).

Distinct

The total number of values in the field, disregarding selections.

Possible

The current number of possible (non-excluded by selections) values in the field.

Comment

Shows an indicator for any special status of the field, such as Semantic, AndMode, AlwaysOneSelected, Info, Locked or Hidden. (See “HidePrefix” and “HidePrefix” on page 345 in Book I).

The list can be sorted on any column by clicking on the column header.

Export Structure

By pressing this button you can export the table structure of the document to a set of text files. These text files, one for the tables (*file-*

name.Tables.tab), one for the fields (*filename.Fields.tab*) and one for mapping in between (*filename.Mappings.tab*) can easily be read back into QlikView for further analysis with the full power of the QlikView logic. A dialog will appear where you can choose the target folder for the export. The default is to put the files in the same folder as the QlikView document.

Tables can be made loosely coupled interactively from this dialog or via macros. There is however also a possibility to declare loosely coupled tables explicitly in the script via the **Loosen Table** script statement (see page 318 in Book I). Tables declared as loosely coupled by such a script statement will behave somewhat differently from those made loosely coupled interactively. The **Loosen Table** script statement will prevent the formation of synthetic keys in the table and the loosely coupled status of that table will not be possible to change interactively via this dialog.

As long as the fields in the document remain the same after script execution, the settings for loosely coupled tables made interactively will survive reload. If any changes occur in the field set as a result of script execution, all settings for loosely coupled tables made before the script execution will be disregarded.

When QlikView finds circular references in the data structure which cannot be broken by tables declared loosely coupled interactively or explicitly in the script one or more additional table will be forced loosely coupled until no circular references remain. When this happens you will be warned by the **Loop Warning** dialog. This will also occur if you attempt to change the loosely coupled setting of a table interactively, thereby creating a circular reference.

Note Making one or more tables loosely coupled can radically change the behavior of your document. Use this feature only when you understand what you are doing!

1.11 Sort

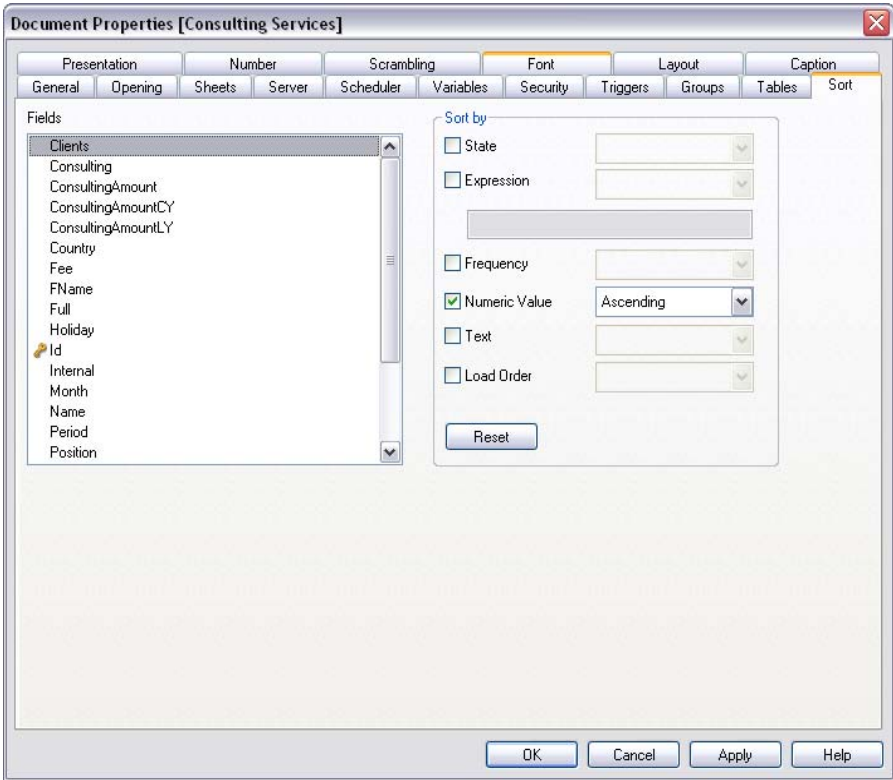


Figure 14. The Sort page in the Document Properties dialog

In this page it is possible to set the default sort order of the fields. The settings are used when creating new sheet objects.

Fields

In this group it is possible to select one or several fields for which the sort order is to be set.

Sort by

In this group, the default sort order of the field values in the sheet objects can be set. Changes made in this group will affect fields in sheet objects created after the change. Previously created sheet objects will not be affected.

State

Sorts the field values according to their logical state (selected, optional, or excluded).

Expression

Sorts the field values according to the expression entered into the text edit box below this sort option.

Frequency

Sorts the field values by frequency (number of occurrences in the table).

Numeric Value

Sorts the field values by their numeric values.

Text

Sorts the field values in alphabetical order.

Load order

Sorts the field values by the initial load order.

Reset

Sets the default sort order.

The order of priority is **State**, **Expression**, **Frequency**, **Numeric Value**, **Text** and **Load Order**. Each of these sort criteria can be set to ascending or descending.

1.12 Presentation

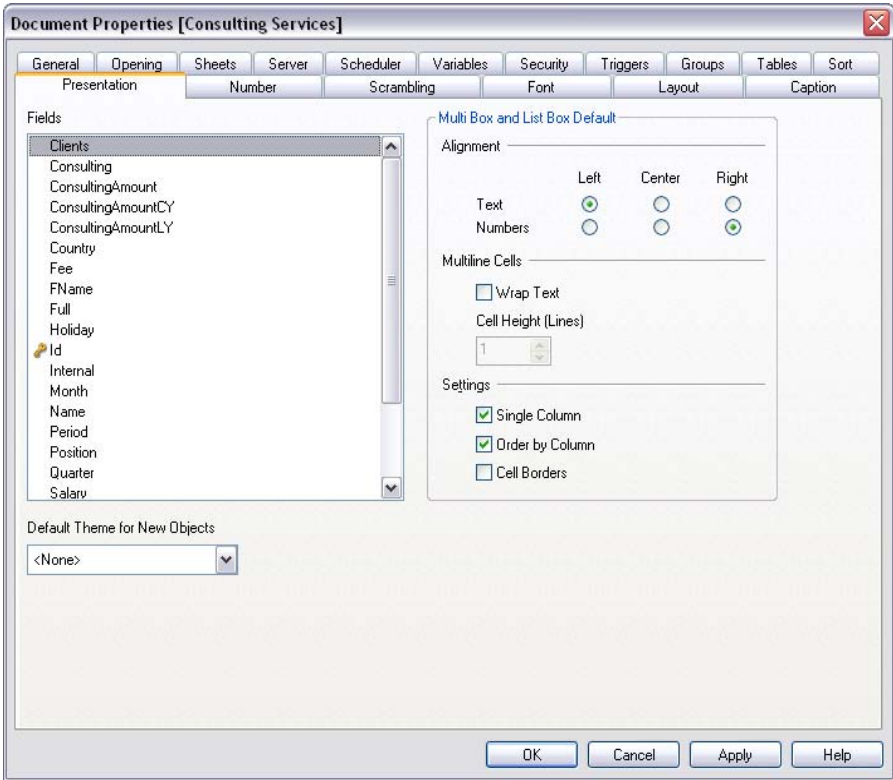


Figure 15. The Presentation page in the Document Properties dialog

In this page it is possible to set the default presentation settings of the field values, which are used when creating new list boxes and when opening multi boxes.

Fields

In this group you select one or several fields for which the different properties are to be set.

Multi Box and List Box Default

Alignment

In this group, the default alignment of your field values can be set. **Text** and **Numbers** can be individually set to left-adjusted, centered or right-adjusted. Changes made in this group will affect fields in sheet objects created after the change. Furthermore, they will affect lists in multi boxes

when these are opened. Previously created sheet objects, however, are not affected.

Multiline Cells

In this group you can set the default display of multiple line cells for the selected fields.

Wrap Text

If this check box is marked, the contents of the cell will be wrapped in two or more lines.

Cell Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the cell should have.

Single Column

Marking this check box will force single column mode.

Order by Column

If this check box is marked, a multiple column presentation will be ordered by columns instead of rows.

Cell Borders

Mark this check box to get horizontal border lines between the different field values in the selected fields.

Default Theme for New Objects

Here you may select a QlikView theme which will be applied to all newly created sheets and sheet objects in the document. The selected theme must be accessible from disc at all times in order to be used. It is also important that the theme used is defined for all types of objects that may occur in a QlikView document. At the bottom of the drop-down list there is a **Browse...** command in case your theme resides in another location than in the default QlikView theme catalog. If no default theme is used, each new sheet object will inherit the properties of the last created or changed object.

1.13 Number

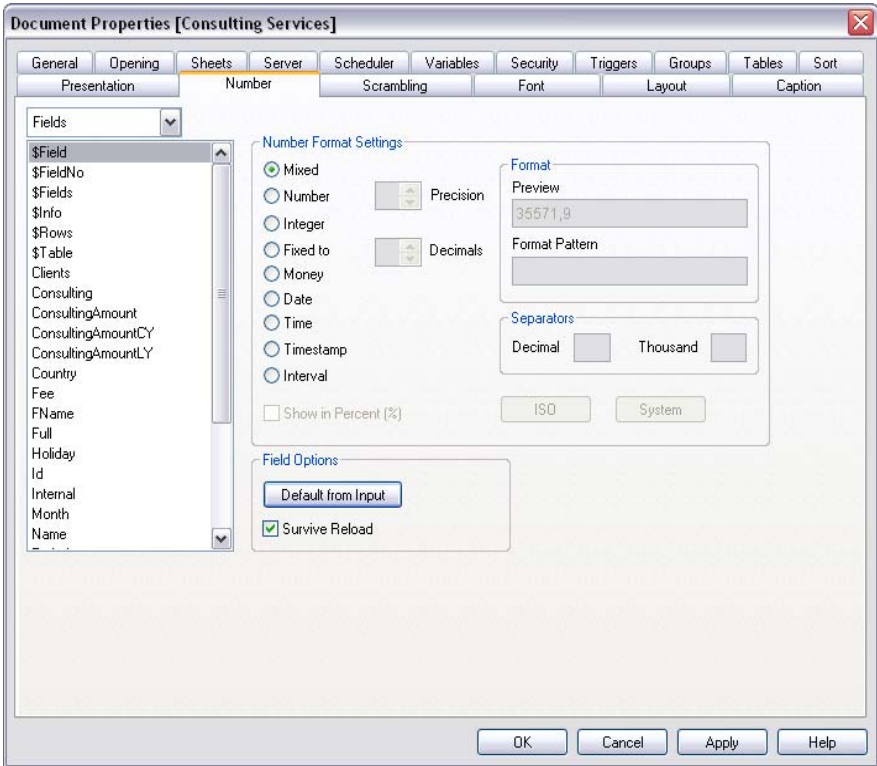


Figure 16. The Number page in the Document Properties dialog

This page is used to set the default number format for fields and variables in the QlikView document.

The list to the left shows either all **Fields** or all **Variables** depending on your selection in the drop-down box above. Select one or more fields/variables and then use the number format controls to set the desired number format.

Some of the number format controls - collected in the **Field Options** group - only apply to fields.

For an explanation of each of the number format controls, see page 301.

1.14 Scrambling

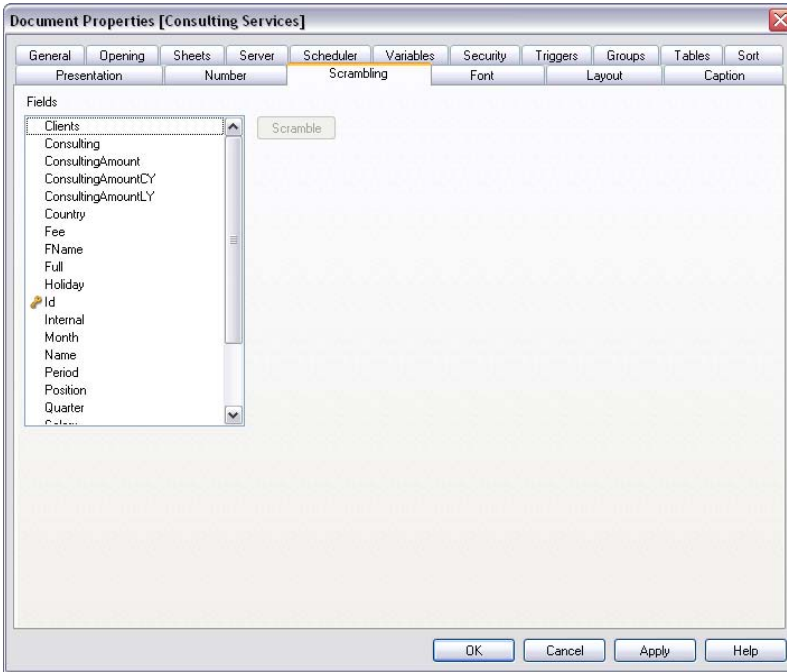


Figure 17. The Scrambling page of the Document Properties dialog.

This page is only available when working with a QlikView document in ADMIN mode. It provides the possibility to scramble the data in one or more fields.

Fields

This is a list of all fields in the document. Select one or more fields for scrambling.

Scramble

Press this button to perform a random scrambling of the contents of selected field(s). Numbers are scrambled to numbers and text to text (spaces are kept). Once scrambled, the data cannot be recreated in its original form by QlikTech or anyone else. If you reexecute the script, the scrambling will however be lost.

1.15 Font

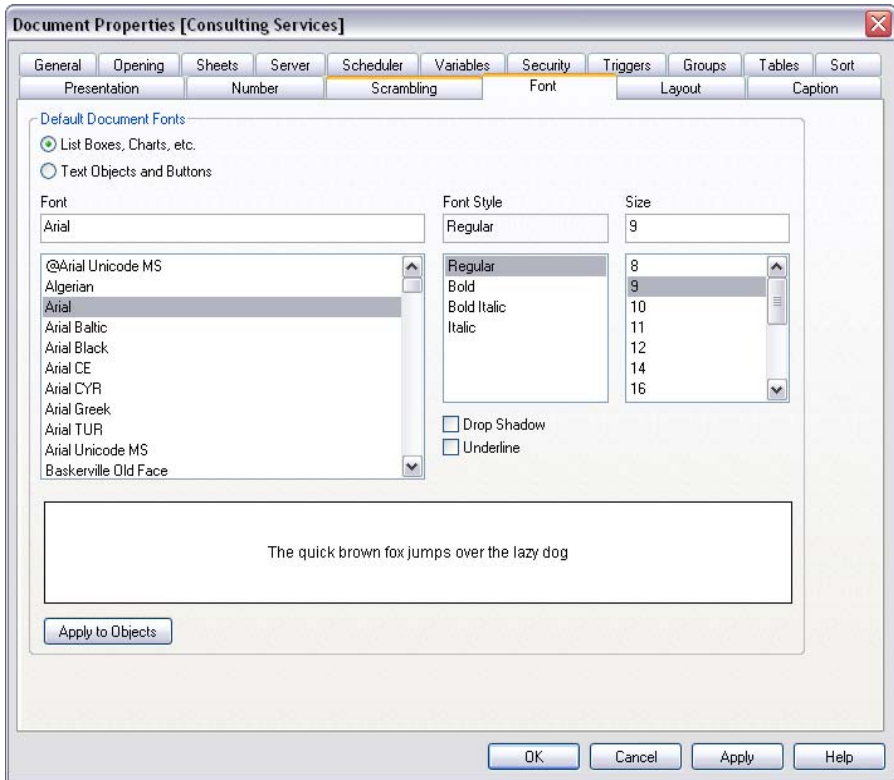


Figure 18. The Font page in the Document Properties dialog

In the **Font** page, you can change one or both of the default fonts of the document. The first default font is used for most objects, including list boxes and charts. The second default font is used for buttons and text boxes, which are objects that usually need a larger font. The fonts chosen will be used for all new objects.

Drop Shadow

If this check box is marked a drop shadow will be added to the text.

Underline

If this check box is marked the text will be underlined.

By using the **Apply to Objects** button, you can also set the font of all the sheet objects in the document at one single time. The font chosen will be applied to all text contained in all sheet objects except where overridden by table styles.

1.16 Layout

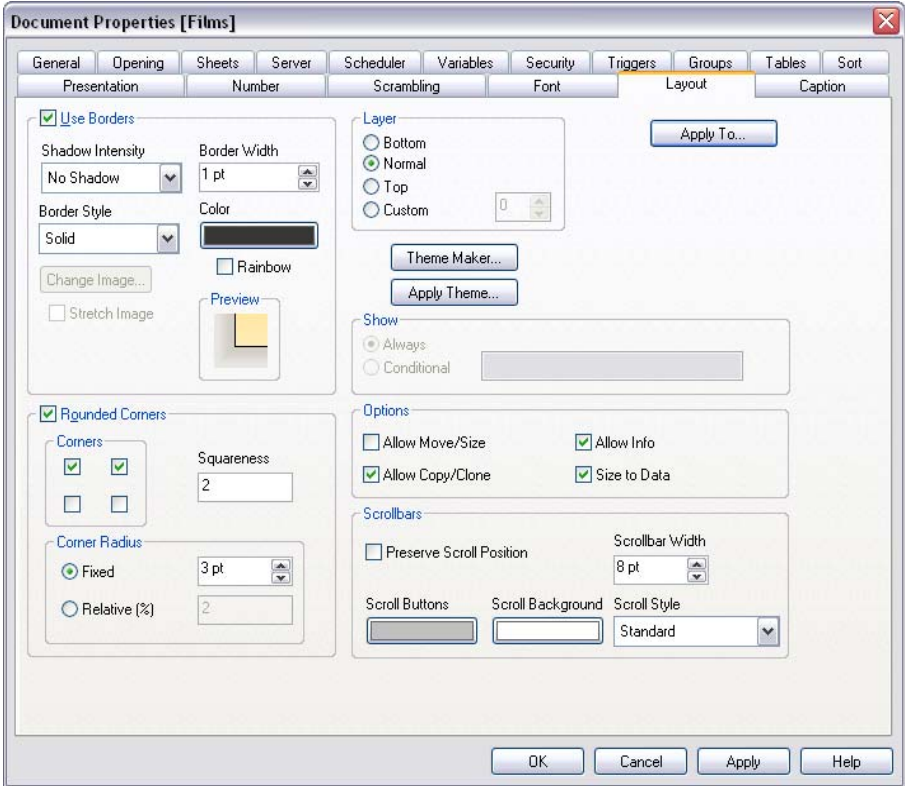


Figure 19. The Layout page in the Document Properties dialog

In the **Layout** page you can specify how the sheet objects should appear on the layout. This includes settings for shape, border and the layer on which they should reside. When changing settings on this page they will apply to all sheet objects in the document.

Use Borders

Mark this check box in order to use a border around the sheet object. You specify the type of border by selecting in the drop-down menu. The **Shadow Intensity** drop-down menu makes it possible to set the intensity of the shadow that surrounds the sheet objects. There is also the choice of **No Shadow**.

Border Width

Here you can set the width of the border. Available for all border types. When setting the width of image borders, make sure the width corresponds to the corner width defined in the image. The width can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Border Style

There are the following predefined border types:

Solid

A solid unicolored border.

Depressed

Border giving the impression of depressing the sheet object from the background.

Raised

Border giving the impression of raising the sheet object from the background.

Walled

Border giving the impression of a wall around the sheet object.

Image

Border specified by custom image.

Change Image

If the **Image** border type is selected, you can click this button to set the image you wish to use.

Stretch Image

If the **Image** border type is used and this check box is marked, the image pattern will be stretched out to fit to the entire space between the corners. If it is not marked, as many copies as possible of the pattern will be displayed, tiled.

Most of the border types can be specified further with the help of the following settings:

Color

Click this button to open a dialog in which you can choose an appropriate base color from the color palette for all border types except **Image**.

Rainbow

Creates a rainbow colored border for all border types except **Image**. The rainbow will start with the selected base color on top of the sheet object.

When **Simplified** is the chosen **Styling Mode**, there is no choice of border type, there is only the **Shadow Intensity** drop-down menu and the **Border Width** setting.

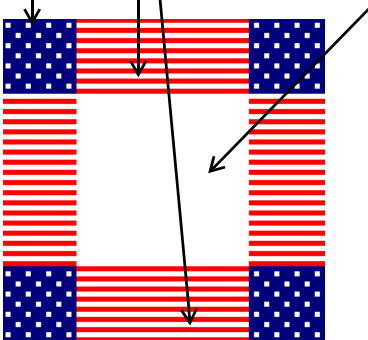
Example of an image border definition file:

An image border is defined by a single image file containing the following elements:

The four corners must form a square of pixels. In the case below each corner is 15 pixels by 15 pixels. The corners may be identical or different, but must all be drawn separately.

The areas between the corners describe the top, bottom, left and right borders between the corners. You only need to define a minimum of pixels which are then extended to fill the space needed by means of stretching or repeating the pattern. See **Stretch image** above.

The area in the middle has no function and may contain anything or nothing.



The number of pixels in the corner square must be correctly stated in the **Width** box above if the border is to come out correctly. It is good practice to name the file in a way that lets the user know about its parameters, e.g. StarsStripes_15_stretch.bmp.

Rounded Corners

In the **Rounded corners** group, settings can be made for the general shape of the sheet object. If the check box is marked, it is possible to use rounded shapes for the corners of the sheet object. It is possible to achieve shapes from rectangular via super elliptic to elliptic/cir-

cular. When **Simplified** is the chosen **Styling Mode**, the **rounded corners** setting is unavailable.

Corners

By marking each of these four check boxes the rounding options are turned on for the respective corners of the sheet object. Corners for which the check box remain unmarked will be drawn rectangular.

Squareness

A number between 2 and 100 setting the general shape of the rounded corners. The number 2 corresponds to a perfect ellipse (or circle for a 1:1 aspect ratio) whereas higher numbers move over super elliptic shapes towards a rectangle.

Corner Radius

This setting determines the radius of the corners in fixed distance or as a percent of the total quadrant. This setting lets you control the extent to which the corners will be affected by the underlying general shape set under **Squareness**. A fixed distance can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Layer

In the **Layer** group, all the sheet objects in the document can be assigned one of three layers on the sheet.

Bottom

A sheet object with the **Bottom** layer property can never obscure sheet objects in the **Normal** and **Top** layers. It can only be placed on top of other sheet objects in the **Bottom** layer.

Normal

When created, sheet objects reside in the normal (middle) layer. A sheet object in the **Normal** layer can never be obscured by sheet objects in the **Bottom** layer and can never obscure sheet objects in the **Top** layer.

Top

A sheet object in the **Top** layer can never be obscured by sheet objects in the **Normal** and **Bottom** layers. Only other sheet objects in the **Top** layer can be placed on top of it.

Custom

The **Top**, **Normal** and **Bottom** layers correspond to internally numbered layers 1, 0 and -1 respectively. In fact all

values between -128 and 127 are accepted. Choose this option to enter a value of your choice.

Apply to...

Opens the **Caption and Border Properties** dialog (see below), where you can set where to apply the properties you set on the **Layout** page.

Theme Maker...

Opens the **Theme Maker** dialog (see page 275) for definition of layout themes.

Apply Theme...

Opens a file browser dialog for opening a QlikView layout theme file to be applied to the document. See page 275 for more information on QlikView layout themes.

Show

In the **Show** group, it is possible to specify a condition under which all the sheet objects are to be shown.

Always

The sheet objects will always be shown.

Conditional

The sheet objects will be shown or hidden depending on a condition expression which will be evaluated each time the sheet objects are to be drawn. The sheet objects will only be visible when the condition returns true. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas (see page 467).

Options

In the **Options** group, it is possible to disallow moving/resizing of the sheet object.


Allow Move/Size

If this check box is deselected it will become impossible for the user to move or resize the sheet object.

Allow Copy/Clone

If this check box is deselected it will become impossible for the user to copy or clone the sheet object.

Allow Info

By default, an info icon  will appear in the top right corner of the sheet object if information is linked to the selected field value. Deselect this check box if you do not

wish the icon to be displayed. This option is available only for list and multi boxes.

Size to Data

Normally the borders around all table sheet objects in QlikView will shrink when selections cause the size of the table to be less than allocated size for the sheet object. By deselecting this check box this automatic adjustment of size will be turned off leaving any surplus space blank.

Scroll Bars

In this group you can set the appearance of scroll bars in the QlikView sheet object.

Preserve Scroll Position

When marking this check box and the corresponding box in **User Preferences, Objects** page, QlikView will try to preserve the scroll position after a selection in table objects and charts with an x-axis scroll bar.

Scroll Background

Sets the scroll bar background color. Select a color by clicking the button. Note that medium gray tones often render the best results for scroll bars.

Scroll Buttons

Sets the scroll button color. Select a color by clicking the button. Note that medium gray tones often render the best results for scroll bars.

Scroll Bar Width

Sets the scroll bar width. The width can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Scroll Style

Sets the scroll bar style. Choose between the **Scroll Styles Classic, Standard and Light**. Scroll Style is not available when **Simplified** is the chosen **Styling Mode**.

Caption and Border Properties



Figure 20. The Caption and Border properties page

In this dialog you set in where in the document your caption and border properties should apply.

Apply Properties to...

Mark this check box to apply your settings to the entire document.

Mark one of the following radio buttons.

Objects on this sheet

Applies your settings to objects on the current sheet only.

Only available when this dialog is opened from the **Layout** page of a sheet object.

Objects in this document

Applies your settings to objects of the entire document.

Mark one of the following radio buttons:

Only this object type

Applies your settings to all objects of this type. Only available when this dialog is opened from the **Layout** page of a sheet object.

All object types

Applies your settings to all objects.

Set as default for new objects in this document

Mark this check box to use your settings as default for all new objects in the current document. Only available when this dialog is opened from the **Layout** page of a sheet object.

1.17 Caption

Caption

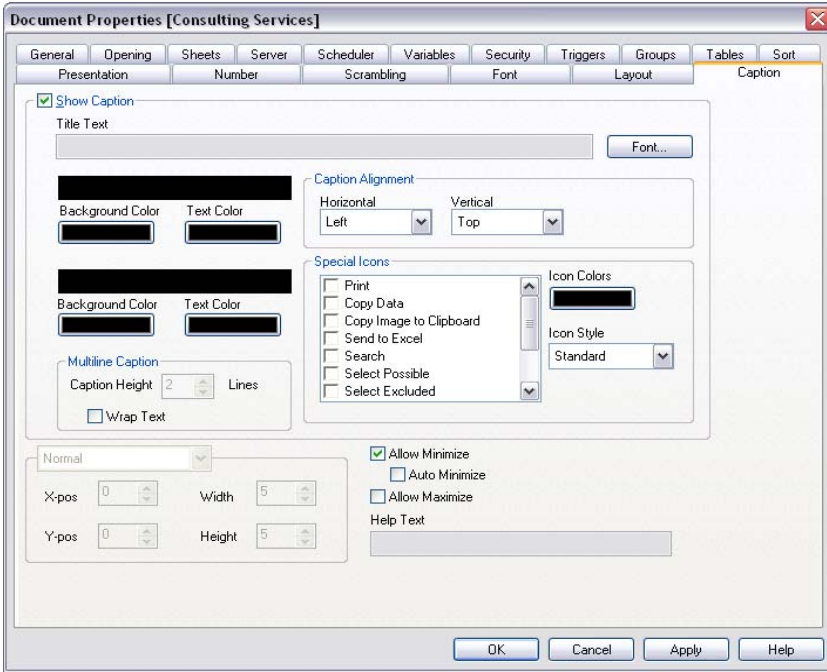


Figure 21. The Caption page in the Document Properties dialog

In the **Caption** page, advanced settings can be made for the caption. Background and foreground text color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction.

Show Caption

If this check box is marked, a caption will be drawn at the top of the sheet object. List boxes and other "box objects" will have caption turned on by default, whereas buttons, text areas and line/arrow objects will have it turned off.

Font

Opens the **Font Dialog** where you may set the font used in the caption.

Title Text

The text to be displayed in the sheet object caption. This setting is identical to the **Windows Title** setting found in the **General** page of the properties dialogs of a number of sheet objects. However, for objects such as the text objects, this is the only place where the caption text can be edited.

Inactive Colors / Active Colors

In this group you can set the colors of the caption in its different states. A preview pane shows the current settings of the selected state.

Background Color

Sets the background color. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the color button.

Text Color

Click the colored button to select a color.

Multiline Caption

In this group you can specify multiple line captions.

Caption Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the caption should have.

Wrap Text

If this check box is marked, the caption will be wrapped in two or more lines.

Normal/Minimized Size/Position

In this group you can set exact pixel size and position of the QlikView sheet object in its normal or minimized state. Select state in the drop-down control.

X-pos

Sets the horizontal position of the left side of the sheet object in relation to the sheet's left edge. The setting is made in pixels.

Y-pos

Sets the vertical position of the top side of the sheet object in relation to the sheet's upper edge. The setting is made in pixels.

Width

Sets the width of the QlikView sheet object in pixels.

Height

Sets the height of the QlikView sheet object in pixels.

Caption Alignment

In this group you set text alignment for caption text.

Horizontal

Sets the horizontal alignment of the caption text. The options are **Left**, **Center** and **Right**.

Vertical

Sets the vertical alignment of the caption text. The options are **Top**, **Center** and **Bottom**. This setting is only relevant when **Multiline Caption** is used.

Special Icons

Many of the object menu commands of the sheet objects can be configured as caption icons. Select commands to be shown as caption icons by marking the check box to the left of each command in the list.

Icon Colors

Sets the background color of the caption icons. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the color button. This setting applies to all icons in the sheet object, not only the special caption icons.

Icon Style

Sets the icon style. Select a style in the drop-down control. This setting applies to all icons in the sheet object, not only the special caption icons.

Note Use the special caption icons with care. One or two of them may provide great help for the end-user while too many may create a “Christmas tree” that severely degrades usability.

Allow Minimize

If this check box is marked, it is possible to turn the sheet object into an icon. The minimize icon will appear in the sheet object caption.

Auto Minimize


This setting, which makes it possible to toggle between sheet objects, should be applied to several sheet objects (preferably positioned on top of each other) on one and the same sheet. Only one of the sheet objects will have its full

size at any given time: as soon as you restore a minimized sheet object with auto minimize set, all other sheet objects with auto minimize set will turn into icons. This option is designed mainly for charts and tables, but may be applied to all sheet objects except buttons, text objects and line/arrow objects.

Allow Maximize

If this check box is marked, it is possible to enlarge the sheet object to fill the sheet. The maximize icon will appear in the sheet object caption.

Help Text

Opens a text box in which you can type an optional help message. The text will be displayed when you position the cursor on the help icon  in the top right corner of the sheet object. The text may be a calculated formula. This option is not available on document level.



2 THE SHEET

A QlikView document can have one or several sheets on which the sheet objects are placed. Each sheet can contain many sheet objects. The sheets have no connection with the logic - if two fields are logically connected, it does not matter if they are put on the same sheet or on different sheets. The logical result when making selections will still be the same.

2.1 Creation

New sheets are created by choosing **Add Sheet** from the **Layout** menu or by clicking the **Add Sheet** button in the toolbar.



2.2 Navigation

All sheets have tabs attached to them on which the sheet name is printed. By clicking on a tab, one activates the sheet attached to it. If the **Sheets** toolbar is active you may also activate a sheet by selecting it in the toolbar drop-down.

By right-clicking on a tab you open a context menu containing the following commands:

Tabrow Properties...

Opens a dialog in which you can choose a font for the tab. Only True-type fonts may be used. See further on page 88.

Sheet Properties...

Selects the sheet and opens the **Sheet Properties** dialog. Here sheet specific parameters such as the sheet background, the fields to display, the default font and the object look etc. can be specified. This alternative is only available when clicking on the tab belonging to the currently active sheet.

Promote (<-) Tab

Moves the tab one step further to the left.

Demote (->) Tab

Moves the tab one step further to the right.

Help

Opens context-specific help.

Remove

Makes the sheet active, then removes it.

If the tab on which you click belongs to a sheet that is currently active, the context menu also contains the commands **Select fields...** and **New sheet object**. For a description of these commands, see below.

On the sheet tabs you can sometimes see selection indicators, small beacons that will guide you to the selections made. These are shown on the tabs of hidden sheets on which you can see selections that are not visible on the active sheet.

2.3 Object Menu for Sheets

By clicking with the right mouse button on the sheet, you open the sheet **Object menu**.

Properties...

Opens the **Sheet Properties** dialog. Here sheet specific parameters such as the sheet background, the fields to display, the default font, the object look etc. can be specified.

Select Fields...

Opens the **Fields** page in the **Sheet Properties** dialog, where you can specify the fields you wish to display as list boxes on the sheet.



New Sheet Object

A menu in which any of the sheet objects can be chosen. A wizard, that helps you create the chosen object, will open. The option **System Table** at the bottom of the menu creates a system pivot table showing the relation of fields and logical tables in the current document.

Copy Sheet

Makes a complete copy of the entire sheet with all its sheet objects. The copy will be named “Copy of *Sheetname*“ and be positioned as the last sheet in the document.

Paste Sheet Object

Pastes a sheet object, previously copied to clipboard on the sheet. Equivalent to the keyboard shortcut CTRL+V.

Print...

Opens the **Print** dialog.

Copy Image to Clipboard

Copies an image of the sheet to the clipboard. The tabrow, the menu bar, the toolbars and the status bar will not be included.

Export Image to File

Opens a dialog for saving an image of the current sheet to file. The image can be saved as bmp, jpg, gif or png.

Help

Opens context-specific help.

Remove

Removes the active sheet.

2.4 Sheet Properties

The sheet properties are set in the **Sheet Properties** dialog, which is opened either by choosing **Properties** from the sheet **Object** menu or by choosing **Sheet Properties** from the **Settings** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).



When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

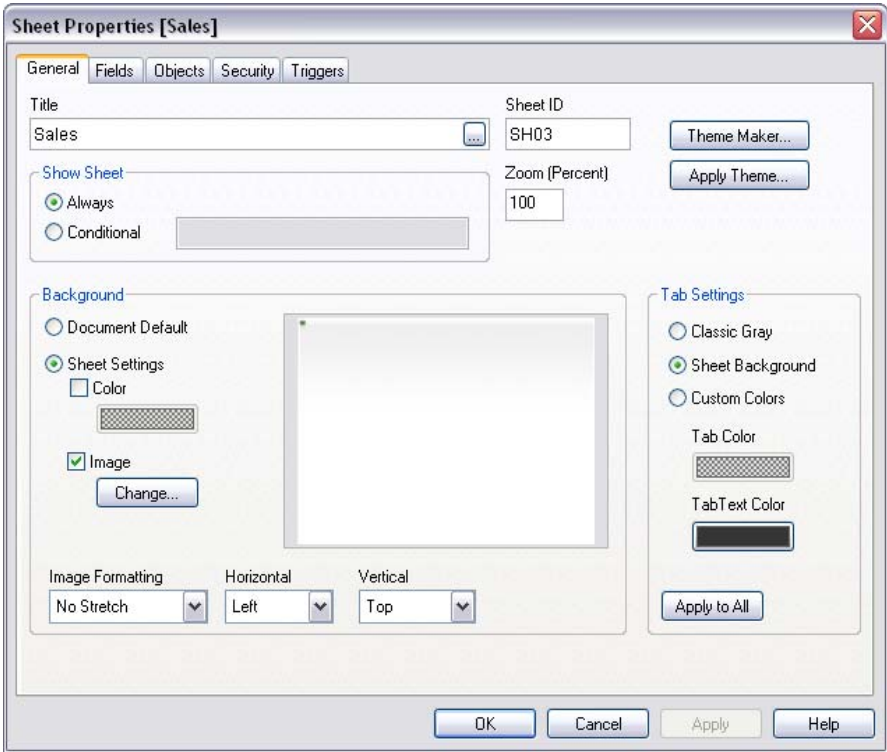


Figure 22. The General page in the Sheet Properties dialog

On the **General** page of the **Sheet Properties** dialog general settings (title, background color etc.) for the sheet are made.

Title

Sets the title for the sheet. This name will be shown on the sheet tab. The title may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Sheet ID

The unique ID of the sheet. Upon creation, every sheet is assigned a unique ID for control via Automation. The first sheet of a document will be assigned the ID SH01. A sheet ID may be changed to any other string that is currently not used as ID for any other sheet, sheet object or bookmark in the document.

Show Sheet

In the **Show Sheet** group, it is possible to specify a condition under which the sheet is shown.

Always

The sheet will always be shown.

Conditional

The sheet will be shown or hidden depending on a condition expression which will be evaluated continuously depending on selections etc. The sheet will only be visible when the condition expression returns true.

When all sheets in a document are disabled due to show conditions, a message "No sheets available" will be displayed. Users with ADMIN privileges for the document can override all show conditions with the **Show All Sheets and Sheet Objects** in the **Security** page of **Document Properties** (see page 42). This functionality can be toggled by pressing CTRL+SHIFT+S.

Zoom

The current zoom factor for the sheet in percent. The value must be an integer between 25 and 400.

Background

In the **Background** group, it is possible to set a background for the sheet.

Document Default

Select this option to use the background set in the **General** page of the **Document Properties** dialog.

Sheet Settings

Select this option to use a separate background for the sheet

Color

Select this option to apply a specific background color to the current sheet. The color can be defined as a solid color or a gradient via the **Color Area** dialog (see page 23) that opens when clicking the button.

Image

Mark this check box to set a background image, a wallpaper, for the sheet. Click the **Change** button to select the wallpaper.

Change

Opens the **Select Wallpaper** dialog, where it is possible to select an image (bmp, jpg, jpeg or png) file to be shown as wallpaper.

Image Formatting

Sets properties for image formatting when image background is used.

No Stretch

If this option is selected, the image will be shown as is, without any stretching. This may cause parts of the picture to be invisible or only part of the sheet to be filled.

Fill

If this option is selected, the image will be stretched to fit the sheet without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the image will be stretched as far as possible to fill the sheet while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the image will be stretched to fill the sheet in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Tile

If this option is selected, the image will be tiled as many times as space allows.

Horizontal

Sets horizontal alignment for image when image background is used.

Vertical

Sets vertical alignment when image background is used.

Theme Maker...

Invokes the **Theme Maker Wizard** for creation of a new theme or editing an existing theme. See page 275 for details on QlikView layout themes.

Apply Theme...

Opens a file browser dialog for opening a QlikView layout theme file to be applied to the sheet object. See “Layout Themes” on page 275.

Tab Settings

In the **Tab Settings** group it is possible to set colors for the sheet tab.

Classic Grey

Select this option to use the classic gray tab coloring with black text from previous versions of QlikView.

Sheet Background

Select this option use the sheet’s background color for the tab.

Custom Colors

Select this option to specify custom colors for tab background and text.

Tab Color

Click this button to select a custom color for the tab background. This option is only available when **Custom Colors** has been selected above.

Tab Text Color

Click this button to select a custom color for the tab text. This option is only available when **Custom Colors** or **Sheet Background** has been selected above.

Apply to All

Click this button to copy the tab color settings of the current sheet to all sheets in the document.

Fields

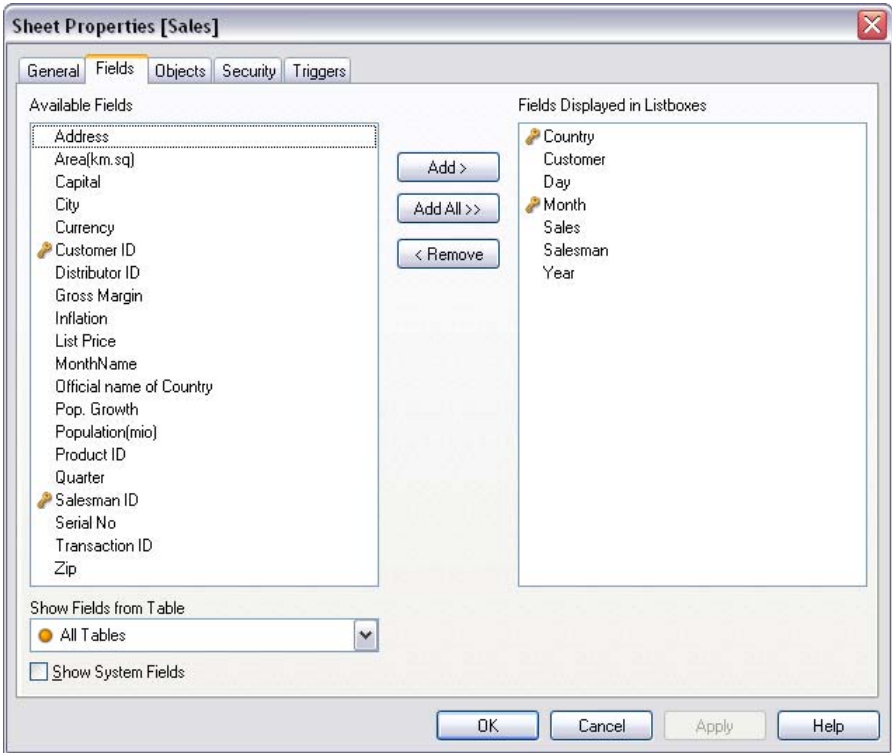


Figure 23. The Fields page in the Sheet Properties dialog

On the **Fields** page of the **Sheet Properties** dialog, you decide what fields to display as list boxes on the sheet.

Available Fields

Available fields are listed in this column. Also the system fields can be listed here.

Show System Fields

Check this box to make the system fields visible in the **Available Fields** column.

Show Fields from Table

Normally this drop-down shows the value **-All Tables-** and the **Available Fields/Groups** list contains all fields and groups of the document. By selecting a specific table name in the drop-down list you may limit the **Available Fields/Groups** list to only fields from that input table.

Finally you may select - **All Tables (Qualified)** - which will show all fields in the document qualified with their table name. A field will appear one time for each table it appears in.

Fields Displayed in List Boxes

Displayed fields are listed in this column. Initially this column is empty.

Add >

Moves the selected fields from the column of available fields to the column of displayed fields.

Add All >>

Moves all the fields in the column of available fields to the column of displayed fields.

< Remove

Moves the selected fields listed in the column of displayed fields to the column of available fields.

Select the fields to be used/removed by clicking on them. Use the **Add >** or **< Remove** buttons to move them to the desired column. You can also move fields by double-clicking them. All the fields shown in both columns are included in the QlikView logic. However, only the ones in the **Fields Displayed in List Boxes** column will be shown on the current tabbed sheet.

In this dialog box it is not possible to select additional fields from the data source. In order to do this, you need to change the script.

Objects page

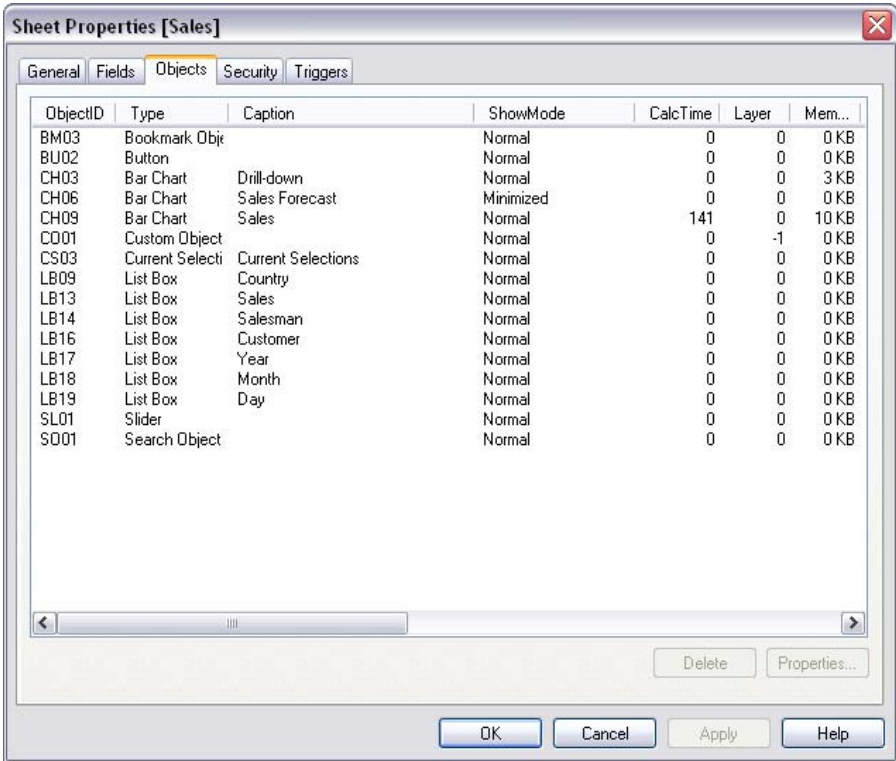


Figure 24. The Objects page in the Sheet properties dialog

This page contains a list of all sheet objects on the current sheet. The columns are as follows:

Object ID

The unique object ID of the sheet object.

Type

The type of the sheet object.

Caption

The text currently shown in the caption of the sheet object.

ShowMode

The current status of the sheet object. Indicates if the object is minimized, maximized, hidden or has a show condition.

CalcTime

The amount of time in milliseconds needed for the last recalculation of the object's contents. This information is only relevant for charts, tables and objects containing one or more formulas.

Layer

The current layout layer of the sheet object. The layers are set on the **Layout** page of sheet object properties dialog (**Bottom**, **Normal**, **Top** and **Custom**), where **Bottom**, **Normal** and **Top** correspond to the numbers -1, 0 and 1. Choose **Custom** to enter a value of your choice. Valid layer numbers are -128 to 127.

Memory

The amount of transient memory needed for the last recalculation of the objects's contents. This information is only relevant for charts, tables and objects containing one or more formulas. The list can be sorted on any column by clicking on the column header. You may also select a specific sheet object by clicking on it in the list. The following two options then become available as buttons below the list:

Left

The current horizontal position of the left edge of the sheet object, expressed in pixels.

Top

The current vertical position of the top edge of the sheet object, expressed in pixels.

Right

The current horizontal position of the right edge of the sheet object, expressed in pixels.

Bottom

The current vertical position of the bottom edge of the sheet object, expressed in pixels.

Width

The current width of the sheet object, expressed in pixels.

Height

The current height of the sheet object, expressed in pixels.

Left (Min)

The current horizontal position of the left edge of the sheet object's minimized icon, expressed in pixels.

Top (Min)

The current vertical position of the top edge of the sheet object's minimized icon, expressed in pixels.

Right (Min)

The current horizontal position of the right edge of the sheet object's minimized icon, expressed in pixels.

Bottom (Min)

The current vertical position of the bottom edge of the sheet object's minimized icon, expressed in pixels.

Width (Min)

The current width of the sheet object's minimized icon, expressed in pixels.

Height (Min)

The current height of the sheet object's minimized icon, expressed in pixels.

The following buttons are activated when a sheet object is selected in the list:

Delete

Deletes the selected sheet object from the sheet. The object immediately disappears from the list but the actual delete will not take place until you select **OK** or **Apply**.

Properties

Opens the **Properties** dialog for the selected sheet object.

Security

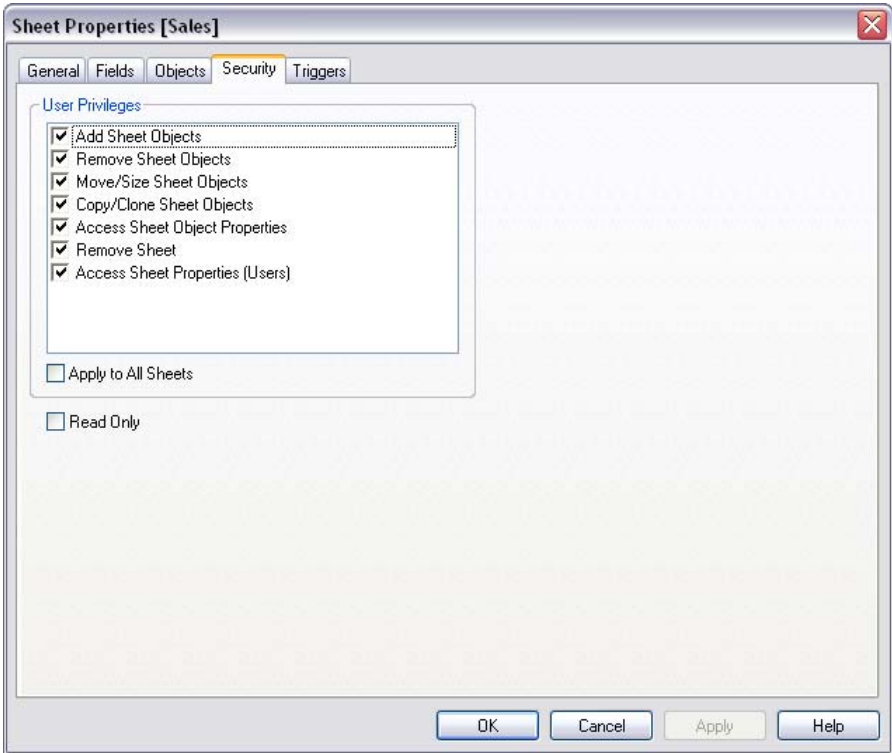


Figure 25. The Security page in the Sheet Properties dialog

This page is useful for documents with access restriction (for more information, see page 505 in Book I). By deselecting an option in the list, the ADMIN user inactivates that particular command or function, thus preventing undesired changes. The Security page is not available for people with USER privileges. By default, all the options are selected, i.e. all the commands are active.

Add Sheet Objects

If this option is deselected, no sheet objects can be added to the current sheet.

Remove Sheet Objects

If this option is deselected, the sheet objects on the current sheet cannot be removed.

Move/Size Sheet Objects

If this option is deselected, the sheet objects on the current sheet cannot be sized nor moved.

Access Sheet Object Properties

If this option is deselected, it is not possible to access the **Properties** dialogs of the sheet objects on the current sheet.

Remove Sheet

If this option is deselected, the current sheet cannot be removed.

Access Sheet Properties (Users)

If this option is deselected, a person with USER privileges cannot access the **Properties** dialog of the current sheet.

Apply to All Sheets

If you mark this check box before choosing **Apply** or **OK**, the options above will be applied to all the sheets of the document.

Read Only

If you mark this check box no changes can be made to the current sheet.

Triggers

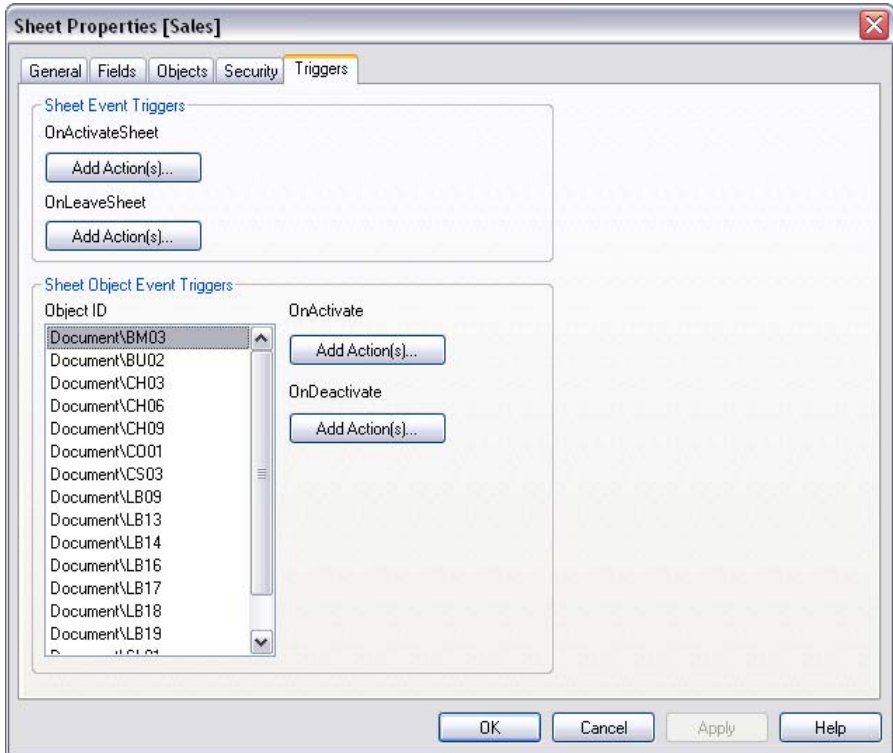


Figure 26. The Triggers page of the Sheet properties dialog

On the **Triggers** page, you can specify actions, including macros, to be invoked on sheet and sheet object events. For more information on macros, see “macros and automation” on page 311.

Not all actions can be invoked from the following event triggers. Read more about the different actions on page 200.

Note There are limitations as to the behavior of certain macro triggers when working with documents on QlikView Server. Please study the special section "Considerations when using macros in QV documents on the QlikView Server" in the QlikView Server reference manual before designing server documents with macro triggers.

Note Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

Sheet Event Triggers

In the **Sheet Event Triggers** group you can set actions to trigger on change of active sheet. Pressing the **Add Action(s)** button for either event brings you to the **Actions** page (see “Actions” on page 200 for how to add actions). On this page you can add an action to the event. Choose the action **macro** to assign an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. When an action has been assigned to the event, the button changes to **Edit Action** and you can change the action for the event. The following **Sheet Event Triggers** are configurable:

OnActivateSheet

The action will be executed each time the sheet is activated.

OnLeaveSheet

The action will be executed each time the sheet is deactivated.

Sheet Object Event Triggers

In the **Sheet Object Event Triggers** group you can set actions to trigger on activation and deactivation of a specified sheet object on the sheet. Select an **Object ID** and choose the event to act as a trigger. Click on the **Add Actions** button to add an action. See “Actions” on page 200 for how to add actions.

OnActivate

The action will be executed each time the sheet object is activated.

OnDeactivate

The action will be executed each time the sheet object is deactivated.

2.5 Tabrow Properties

The properties of the tabrow are set in the **Tabrow Properties** dialog, which is opened by right-clicking the tabrow.

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

Font

In the **Font** page (see Figure 33 on page 113), you can change the font of the text on the tab.

You can specify the font, style and size of the text. The font chosen applies to the text on all the tabs.



3 SHEET OBJECTS

There are fourteen types of sheet objects: list boxes, statistics boxes, multi boxes, table boxes, charts, input boxes, current selections boxes, buttons, text objects, line/arrow objects, slider/calendar objects, custom objects, search objects and bookmark objects.

3.1 Local Objects and Server Objects

There are three different types of sheet objects:

Document Sheet Objects are stored inside the qvw document. They will always be available to whoever opens the document locally or from QlikView Server.

Personal Server Objects are only available when working with a document on QlikView Server and only to authenticated users. They are stored in a repository on the server and are accessible to the user from any computer where authenticated. Your personal server objects can be managed from the **My Server Objects** page of the **Server Objects** dialog accessible from the **Layout** menu (see below) or through the **Collaboration Pane**, accessible from the **View** menu.

Shared Server Objects are only available when working with a document on QlikView Server and only to authenticated users. Any user who has created a personal server object may flag this as shared. They will then become available to other users. Just like personal server objects, shared server sheet objects are stored in a repository on the server. Available shared server objects can be managed from the **Shared Server Objects** page of the **Server Objects** dialog that is accessible from the **Layout** menu (see below) or through the **Collaboration Pane**, accessible from the **View** menu.

3.2 Creation

New sheet objects can be created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu. Another alternative is to choose the button corresponding to the respective sheet object in the toolbar.

3.3 Navigation

Active Object

There is normally one sheet object active. It is normally the one with a differently colored header (the active title bar color). You make a sheet object active by clicking on it. The **TAB** (**SHIFT+TAB**) key makes the next (previous) sheet object active. Also objects without caption, such as buttons can be active. Pressing **F1** gives help on the object.

Activating Sheet Objects

You can activate more than one sheet object at the same time by holding down the **SHIFT** key while clicking on the headers of the objects. It is also possible to draw a rectangle around a number of sheet objects with the left mouse button depressed. All sheet objects fully surrounded by the rectangle will be activated. Finally, all sheet objects on the active sheet can be activated via the keyboard shortcut **CTRL+A** or the command **Activate All** on the **Edit** menu. Multiple active sheet objects can be moved, duplicated, aligned, sized, and removed as a group. Furthermore, it is possible to make a text search in all the active objects at the same time. Read more about text search on page 131 in Book I.

3.4 Object Menu for Sheet Objects

The **Object** menu for the sheet object can be opened from the menu bar, by clicking with the right mouse button on the sheet object.

Under some circumstances, e.g. when a sheet does not yet contain any sheet objects, the **Object** menu for the sheet is found as **Object** menu in the menu bar.

When no sheet object is active, the **Object** menu in the menu bar does not open. If there is more than one active sheet object, the **Object** menu contains the commands common to the active objects.

3.5 Sheet Object Properties

Each sheet object has a dialog where its properties can be set. The dialog can be opened by choosing **Properties** from the **Object** menu for the sheet object. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

3.6 Minimized and Maximized Sheet Objects

All types of sheet objects have the options of being minimized to an icon or maximized to fill the whole sheet. These options can be individually turned on and off in the **Caption** page of the sheet object's **Properties** dialog. The default status of the settings varies with the type of object.

Changing the status of an object between normal, minimized and maximized mode can be done by clicking on icons in the sheet object caption, by double-clicking on the caption and for most sheet objects via commands on the object's context menu.

Minimized sheet object icons can be freely positioned on the sheet and sized within certain limits.

3.7 The Server Objects Dialog

The Server Objects dialog offers another more list-oriented way to manage your personal server objects and to select among other users' shared server objects. The dialog contains two pages:

My Server Objects

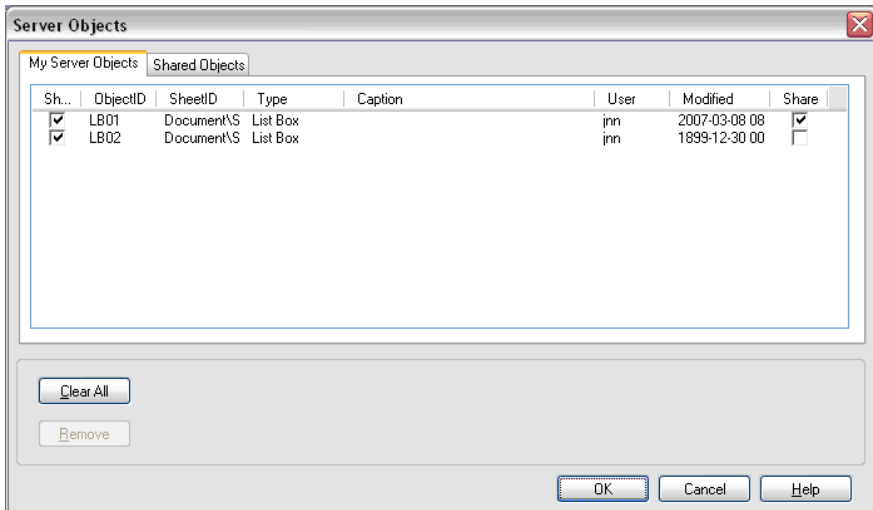


Figure 27. The My Server Objects page of the Server Objects dialog

In the **My Server Objects** page you can manage your own personal server objects and share them with other users.

At the top of the dialog you will find a list of all personal server objects currently defined for the QlikView document. The objects are described by, and may be sorted by several columns:

Show

Mark this check box if you want the object to appear in the layout of the document. If the check box is left unchecked, the object will not appear, but will remain accessible via this dialog.

Object ID

The unique ID of the object.

Sheet ID

The ID of the sheet where the object will reside when shown.

Type

The type of the object (e.g. list box).

Caption

The caption of the object (if defined).

User

The Windows identity of the user who created the object (yourself).

Modified

A timestamp indicating when the object was last modified.

Share

This check box is only available in the **My Server Objects** page. By marking it for one of your personal server objects, you will make that object available to other authenticated users of the same server document in their **Shared Objects** page. The object will stay in your **My Server Objects** page and will not appear in your **Shared Objects** page. You may at any time revoke the sharing by deselecting the check box.

At the bottom of the dialog you will find buttons which can be used to perform actions with the objects in the list above.

Clear All

Removes all objects in the list.

Remove

Removes the selected object from the list.

Shared Objects

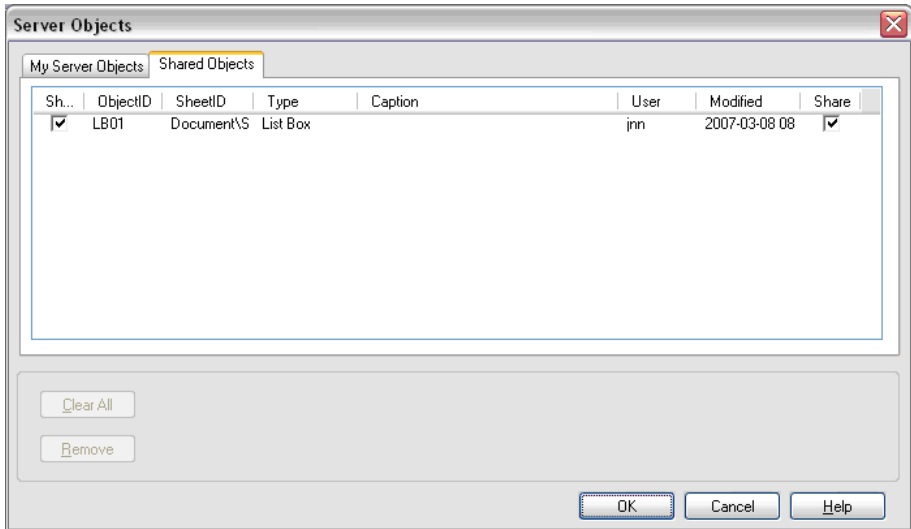


Figure 28. The Shared Objects page of the Server Objects dialog

In the **Shared Objects** page you can manage the display of other users' shared server objects.

At the top of the dialog you will find a list of all shared server objects currently defined for the QlikView document. The objects are described by, and may be sorted by several columns:

Show

Mark this check box if you want the object to appear in the layout of the document. If the check box is left unchecked, the object will not appear, but will remain accessible via this dialog.

Object ID

The unique ID of the object.

Sheet ID

The ID of the sheet where the object will reside when shown.

Type

The type of the object (e.g. list box).

Caption

The caption of the object (if defined).

User

The Windows identity of the user who created the object.

Modified

A timestamp indicating when the object was last modified.

Share

This check box is only available in the **My Server Objects** page. By marking it for one of your personal server objects, you will make that object available to other authenticated users of the same server document in their **Shared Objects** page. The object will stay in your **My Server Objects** page and will not appear in your **Shared Objects** page. You may at any time revoke the sharing by deselecting the check box.

At the bottom of the dialog you will find buttons which can be used to perform actions with the objects in the list above.

Clear All

Removes all objects in the list.

Remove

Removes the selected object from the list.

Sheet Object Drag and Drop to Microsoft Office

Sheet objects in QlikView can now be dragged and dropped straight into Microsoft Word, Excel and PowerPoint documents. This feature works for local and server QlikView documents and from installed QlikView and QlikView IE plug-in alike. The objects retain their interactivity, which means it is possible to do live QlikView analysis in e.g. PowerPoint. Simply select an object and drag it to the Office application to use it outside QlikView.

Note The QlikView IE plug-in must be installed on the computer where the Microsoft Office document is used.

4 THE LIST BOX

This is the most basic object on the screen. It contains a list of all the possible values of a specific field. In a list box you often make selections, and look at logical connections and implications.

A list box may also contain a cyclic or drill-down group. If a drill-down group is used, selecting a single value in the list box will cause the list box to drill down and switch to the next underlying field in the group. Drilling back up can be accomplished by clicking the drill-up icon in the list box caption..

If a cyclic group is assigned to a list box it can be made to show the next field in the group by clicking on the cycle icon that appears in the list box caption. You may also go directly to any field in the group by using the **Cycle->** option on the list box object menu.

Country	
Argentina	3
Austria	2
Belgium	2
Brazil	9
Canada	3
Denmark	2
Finland	2
France	11

4.1 Creation

The first thing to do after executing the script or after creating a new tabbed sheet is to select fields to be displayed as list boxes. This is done from the **Fields** page of the **Sheet Properties** dialog. Some or all of the fields retrieved from the database can be displayed as list boxes.

New list boxes can also be created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu, or by clicking the **Create List Box** button in the toolbar.



4.2 List Box Object Menu

The list box Object menu is found as the Object menu when a list box is active. It can also be opened as a context menu by clicking with the right mouse button on a list box. The menu contains the following commands:

Properties...

Opens the **List Box Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Search

Opens the text search window. The action is equal to that of just start typing in an active list box. The search window will contain two wildcard characters (**) with the cursor placed in-between. As

you type, all values which have an exact match with the search text will be shown in the list. If you hit ENTER all matching values in the list will be selected. This command can also be invoked via the keyboard shortcut CTRL+F.

Fuzzy Search

Opens the search window in fuzzy search mode. The search window will contain a tilde ~ with the cursor placed after it. As you type, all values will be sorted by the degree of resemblance to the search string with the best matches at the top of the list. If you hit ENTER the first value in the list will be selected.

Advanced Search

Opens the **Advanced Search** dialog. The dialog makes it possible to enter advanced search expressions. This command can also be invoked via the keyboard shortcut SHIFT+CTRL+F.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Change Value

Only available for list boxes displaying input fields. Sets the cell that has been clicked in input edit mode. Equivalent to clicking the input icon in the cell.

Restore Values

Only available for list boxes displaying input fields. Opens a menu with the following options.

Restore Single Value

Restores the value in the field value that has been clicked to its default value from the script.

Restore Possible Values

Restores the values of all possible field values to their default values from the script.

Restore All Values

Restores the values of all field values to their default values from the script.

Select Possible

Selects the possible (selected and optional) field values in the list box.

Select Excluded

Selects the currently excluded field values in the list box.

Select All

Selects all the field values in the list box.

Clear

Deselects all the values in the list box. This command can also be invoked via the keyboard shortcut CTRL+D.

Clear Other Fields

Selects all possible values in the current list box, then clears the selections in all the other fields.

Lock

Locks the selected values in the list box. This command can also be invoked via the keyboard shortcut CTRL+L.

Unlock

Unlocks the locked values in the list box. This command can also be invoked via the keyboard shortcut CTRL+U.

Create QuickChart...

Opens the **QuickChart wizard**, which helps you to create a one-dimensional chart in a quick and easy way.

Create Time Chart

Opens the **Time Chart wizard** to help you build charts where a given measure (expression) should be qualified and often compared by different time periods, e.g. current year, last year, year-to-date etc.

Create Statistics Box

When this command is chosen, a statistics box is opened. It shows selected statistical properties of the field, and is updated dynamically.

Print Possible...

Opens the standard **Print** dialog box (see page 263), allowing you to print the possible (selected and optional) values of the current list box.

Print possible as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports possible (including selected) values in the list box to Microsoft Excel. The values will appear as one column in a new Excel worksheet. For this functionality to work, Microsoft Excel 97 or later must be installed on the computer.

Export...

Opens a dialog where you can export the possible values in the list box to a file of your choice. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files, see page 497 in Book I).

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Possible Values

Copies the possible (selected and optional) values to the clipboard.

Cell Value

Copies the text value of the list box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.

Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Cycle->

Only available when the list box is set to display a cyclic group. Opens a menu with the fields in the group for direct selection of which field to show in the list box.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the current list box from the sheet.

4.3 List Box Properties

The **List Box Properties** dialog box is opened by selecting **Properties** from the list box **Object** menu or using ALT+RETURN. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **List Box Properties** dialog box contains several pages:

General

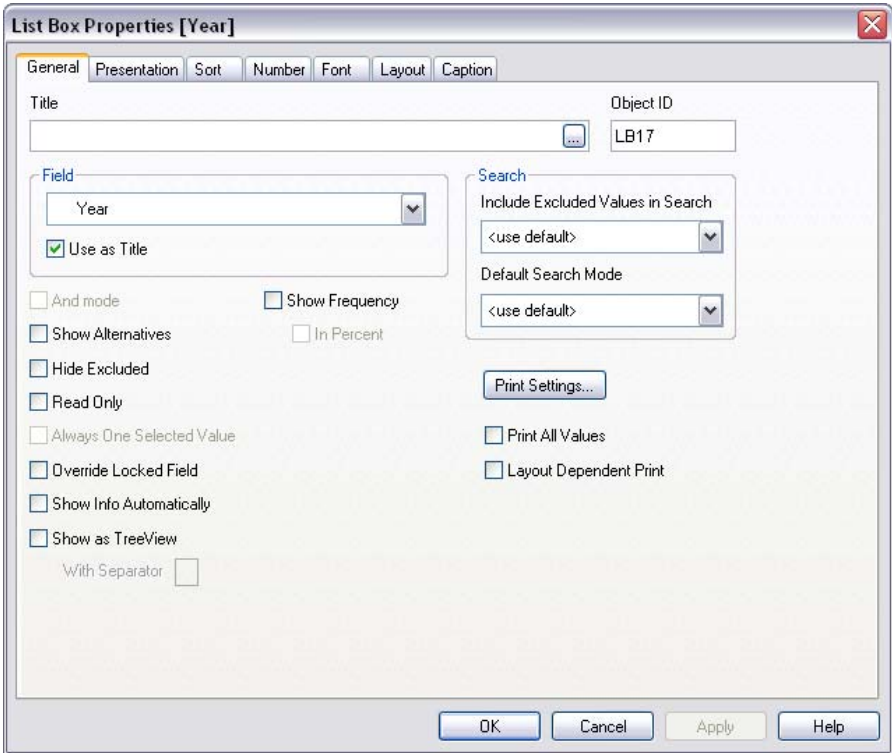


Figure 29. The General page in the List Box Properties dialog

The **General** page contains some general settings for the list box:

Title

An alternative name for the list box can be entered here. It will be used as window title for the sheet object. Note that this name does not affect the logical evaluation based on the field names. The title may be defined as a calculated formula (see page 467) for dynamic update. If no title is stated, the name of the field shown in the list box will be used. This is indicated in the edit box as <use field name>.

Field

The field to be shown in the list box. You may also select a cyclic or drill-down group for the list box. Only one field will be shown at a time though. Furthermore it is possible to display a calculated field in a list box. A calculated field does not come out of script execu-

tion, but is defined from an expression. The same rules as with calculated dimensions in charts apply (see chapter 19 on page 293 in Book III). If you want to use a calculated field in the list box, select <Expression> in the drop-down. This will open the **Edit Expression** dialog where you define the expression that describes the calculated field.

Use as Title

By selecting this check box, the field name will be used as **Title**. The text <use field name> will appear in the **Title** edit box.

Edit...

This button is only available when <Expression> has been selected as field name. The button opens the Edit Expression dialog where the expression describing a calculated field can be defined.

Note When a calculated field is used some list box functionality will be different as compared to regular field list boxes. Some property options are not available (see below). Furthermore statistics boxes cannot be created from such list boxes.

Note A calculated field must always be a transformation of one or more normal fields. Purely synthetic calculated fields created via the **valuelist** and **valueloop** functions are not permitted. When making selections in a calculated field, you actually make selections in the underlying normal field(s). When using more than one field as base for a calculated field, the logical implications of selections in the base fields or the calculated field may be quite complex.

Note The use of calculated fields in list boxes may cause performance issues with very large data sets.

And mode

A multiple selection in a list box must be either of an *and* or an *or* type. By default the selection is of an *or* type. This switch toggles the logical mode in which multiple value selections are made. This menu option is disabled if the field does not satisfy certain conditions (see page 137 in Book I). This option is not available for calculated fields.

Show Alternatives

Toggles whether or not a selection in a list box should directly make the other values listed in the same list box gray. If the **Show Alternatives** option is on, the color of all the other values will be affected only by selections in other list boxes, even though they are logically excluded by the selection in the same list box. If the **Show Alternatives** option is off, all the other values will be gray. This option is not available for calculated fields.

Hide Excluded

Toggles whether or not the excluded values of the field should be shown. If this option is used, excluded values also become unselectable in the list box.

Read Only

Marking this check box makes selections in the list box impossible.

Always One Selected Value

If this check box is marked, the list box will always have one and only one value selected. This is useful when creating applications where one value always has to be selected in order for charts etc. to make sense. An example could be the selection of one currency rate. When this option is marked, it is not possible to select an excluded value in another list box. The option is available only when you have already selected exactly one value in the list box. This option is not available for calculated fields.

Override Locked Field

If this check box is marked, selections can be made in the field from the list box even if the field is locked. The field will still be locked for logical changes stemming from selections in other fields.

Show Info Automatically

Toggles whether or not associated information should be shown automatically when a value is selected. If it is not shown automatically, the info symbol must be clicked.

Show as Tree View

Shows the list box as a tree view (see page 123). This control is only relevant if the field contains the path representation of nodes in a hierarchy. Such a field can be generated using the *Path* parameter of the *Hierarchy* prefix.

Separator

Sets the character that should be interpreted as separator in the path used for the **Tree View**.

Show Frequency

Toggles the status as to whether or not the frequency of a field value is shown. Frequency is here the number of records containing possible values of the current field. This option is normally not available for fields used for associations between tables (see page 464 in Book I). This option is not available for calculated fields.

In Percent

Toggles the status as to whether or not the frequency should be shown in absolute numbers or as percentages of the total number of entries.

Object ID

The unique ID of the current list box. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first list box of a document will be assigned the ID LB01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Include Excluded Values in Search

Specifies whether excluded values should be included in text searches. Three alternatives are given.

<use default>

The default as specified under **User Preferences** will apply. (See page 105 in Book I).

Yes

Excluded values are always included in the text search.

No

Excluded values are not included in text search. Only optional values are included.

Default Search Mode

Specifies the default search mode to be used in text searches. The mode can always be intermittently changed by typing * or ~ as part of the search string. Three alternatives are given.

<use default>

The default as specified under **User Preferences** will apply. (See page 95 in Book I).

Use Fuzzy Search

Fuzzy search is default when starting a text search.

No

Normal search is default when starting a text search.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Print All Values

The normal print behavior for list boxes is to print only possible values. By marking this check box, all values (also excluded) will be printed.

Layout Dependent Print

By marking this check box, the list box will be printed as it appears on the screen in terms of multiple columns, selection color coding of cells etc. This may be useful e.g. when including list boxes in reports.

Presentation

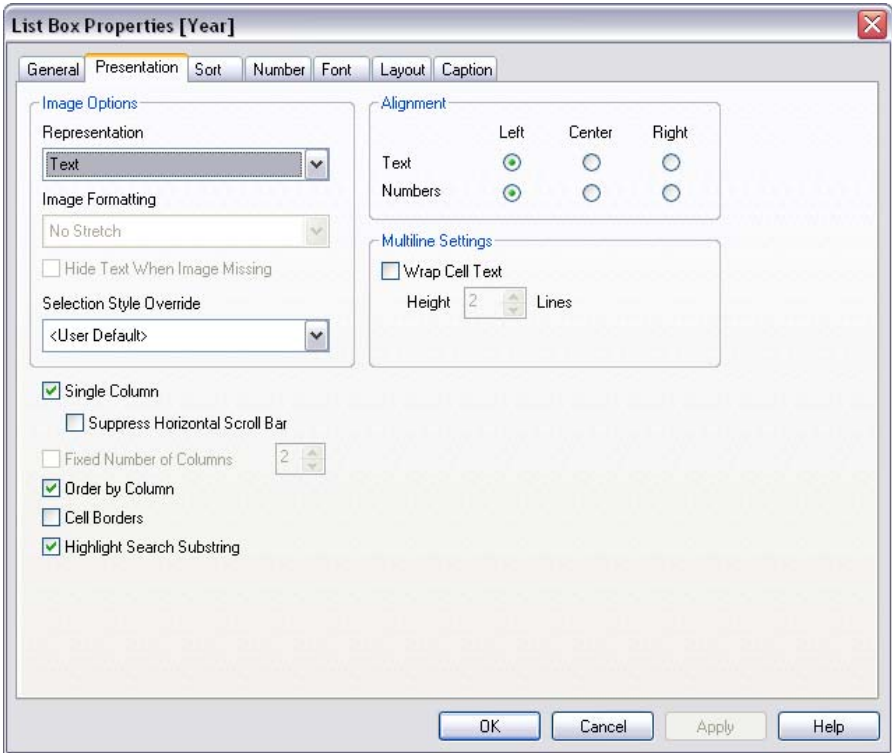


Figure 30. The Presentation page in the List Box Properties

The **Presentation** page contains layout options, e.g. the alignment.

Image Options

This group allows you to have a field value interpreted as a reference to an image in memory or on disk. The following settings are possible:

Representation

Three alternatives are given.

Text

When selecting this option the field values will always be interpreted and displayed as text.

Image

When selecting this option QlikView will try to interpret each field value as a reference to an image. The reference may be a path to an image

file on disk (e.g. `c:\MyPic.jpg`) or inside the qvw document (e.g. `qmem://<Name>/<Peter>`). If QlikView cannot interpret a field value as a valid image reference, the value itself will be displayed.

Info as Image

When selecting this option QlikView will display image info linked to the field value via `info load/select` in the script. If no image is available for the field value, the value itself will be displayed.

Image Formatting

Only available when one of the image options has been selected above. This setting describes how QlikView formats the image to fit in the cell. There are four alternatives.

No Stretch

If this option is selected, the image will be shown as is, without any stretching. This may cause parts of the picture to be invisible or only part of the cell to be filled.

Fill

If this option is selected, the image will be stretched to fit the cell without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the image will be stretched as far as possible to fill the cell while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the image will be stretched to fill the cell in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Hide Text When Image Missing

If this option is selected, QlikView will not display the field value text if interpretation as an image reference fails for some reason. The cell will then be left blank.

Selection Style Override

When images are shown instead of text it may be necessary to use another selection style than the document default so that the logical state of the field values becomes visible.

Use the drop-down to select a suitable selection style, e.g. Corner Tags. This option is only available when image representation of some form has been selected under **Representation**.

Alignment

In this group, the alignment within the list box can be set. **Text** and **Numbers** can be individually set to left-adjusted, centered or right-adjusted.

Multiline Settings

In this group you can set the default display of multiple line cells.

Wrap Cell Text

If this check box is marked, the contents of the cell will be wrapped in two or more lines.

Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the cell should have.

Single Column

Marking this check box will force single column mode. If it is not marked, you will automatically get several columns when sizing it by dragging its border.

Suppress Horizontal Scroll Bar

Mark this check box to suppress horizontal scroll bar in list box when field values are too wide to be displayed in their entirety.

Fixed Number of Columns

If this check box is marked you may select a fixed number of columns for the list box. The item is only available when **Single Column** is deselected.

Order by Column

If this check box is marked, a multiple column list box will be ordered by columns instead of rows.

Cell Borders

Mark this check box to get horizontal border lines between the different field values. Cell borders are automatically turned on when **Wrap Text** is selected, but can subsequently be turned off.

Highlight Search Substring

The search string entered in a search object so far will be highlighted in the matches.

Sort

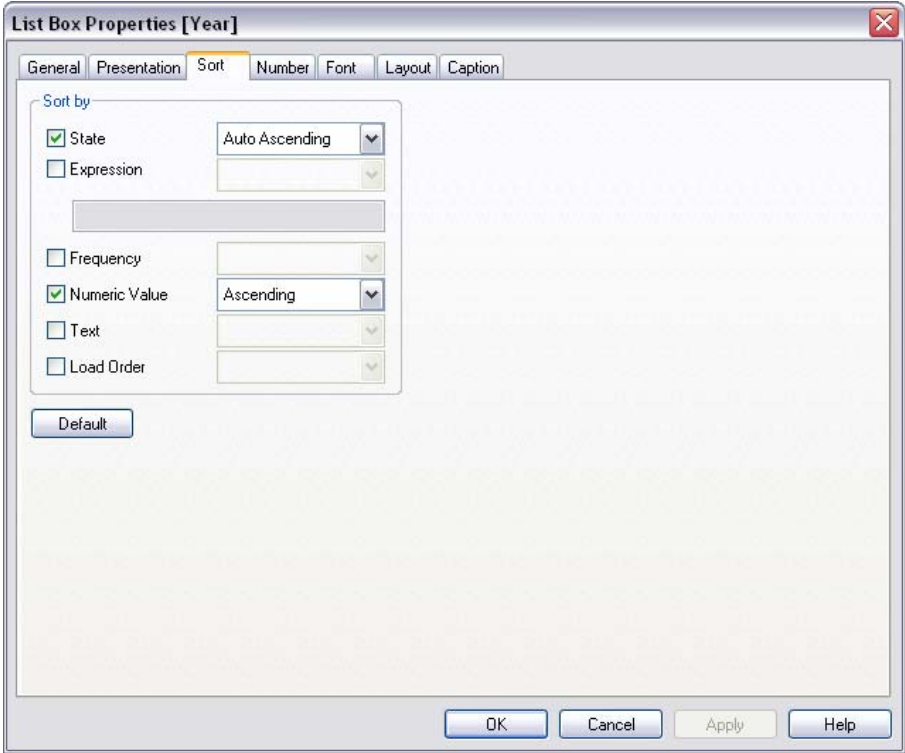


Figure 31. The Sort page in the List Box Properties dialog

The third page, the **Sort** page, contains options to set the sort order within the list box:

Sort by

In this group, the sort order of the field values in the list box can be set.

State

Sorts the field values according to their logical state (selected, optional or excluded).

If **Auto Ascending** is set and the list box is large enough to display all values, the list box will be sorted according to the second sort order set, **Ascending** if no other sort order is defined. If not all values can be displayed in the list box, the sort order will be **Ascending**.

Expression

Sorts the field values according to the expression entered into the text edit box below this sort option.

Frequency

Sorts the field values by frequency (number of occurrences in the table).

Numeric Value

Sorts the field values by their numeric values.

Text

Sorts the field values in alphabetical order.

Load order

Sorts the field values by the initial load order.

Default

Sets the default sort order.

The order of priority is **State, Expression, Frequency, Numeric Value, Text** and **Load Order**. Each of these sort criteria can be set to ascending or descending.

Number

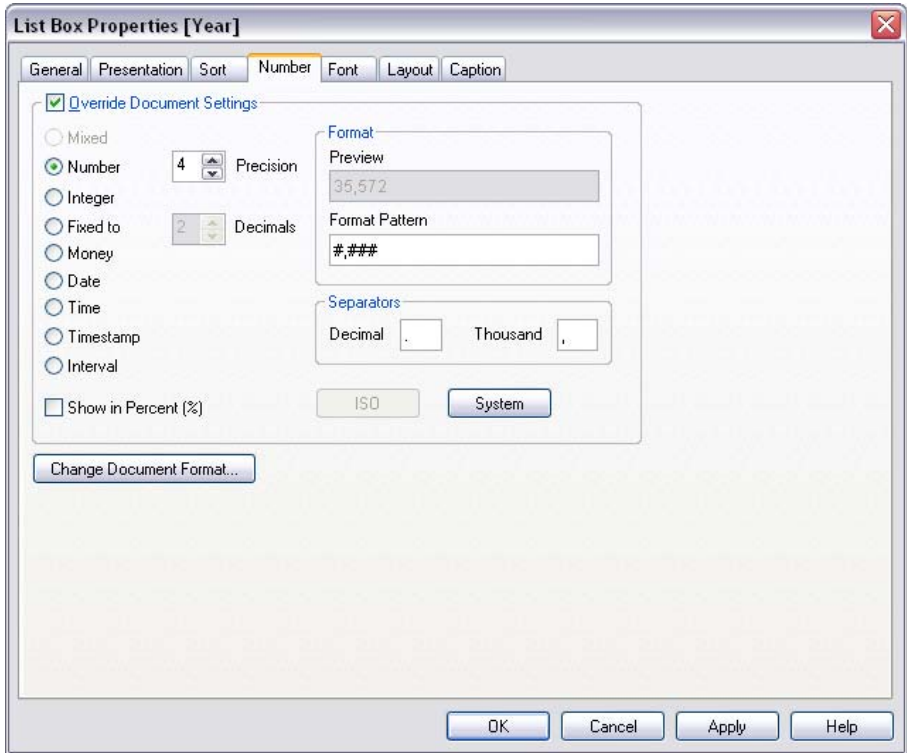


Figure 32. The Number page in the List Box Properties dialog

In the **Number** page, it is possible to set the number format for the list box.

Override Document Settings

Each field has a number format which can be set on the **Number** page of the **Document Properties** dialog. Normally this is the number used when displaying field values in a list box. It is however possible to use a separate number format in an individual list box. To do this, mark this check box and specify a number format in the group control below. For an explanation of the controls, see the number format section, page 301.

Change Document Format

Opens the **Number** page of the **Document Properties** dialog, where you can change the underlying number format in the field.

Font

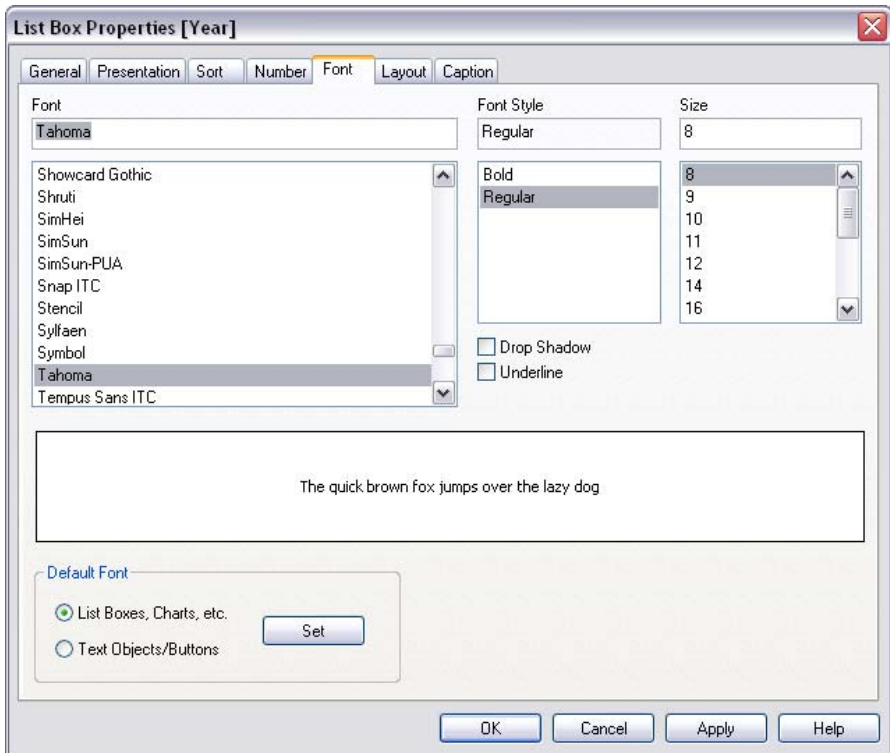


Figure 33. The Font page in the List Box Properties dialog

In the **Font** page, you can change the font of the text in the list box.

The font chosen will be applied to all text contained in the list box. You can specify the font, style and size of the text. Additional settings are:

Drop Shadow

If this check box is marked a drop shadow will be added to the text.

Underline

If this check box is marked the text will be underlined.

By using the **Set** button under **Default Font**, you can also set the default font for the object type.

Layout

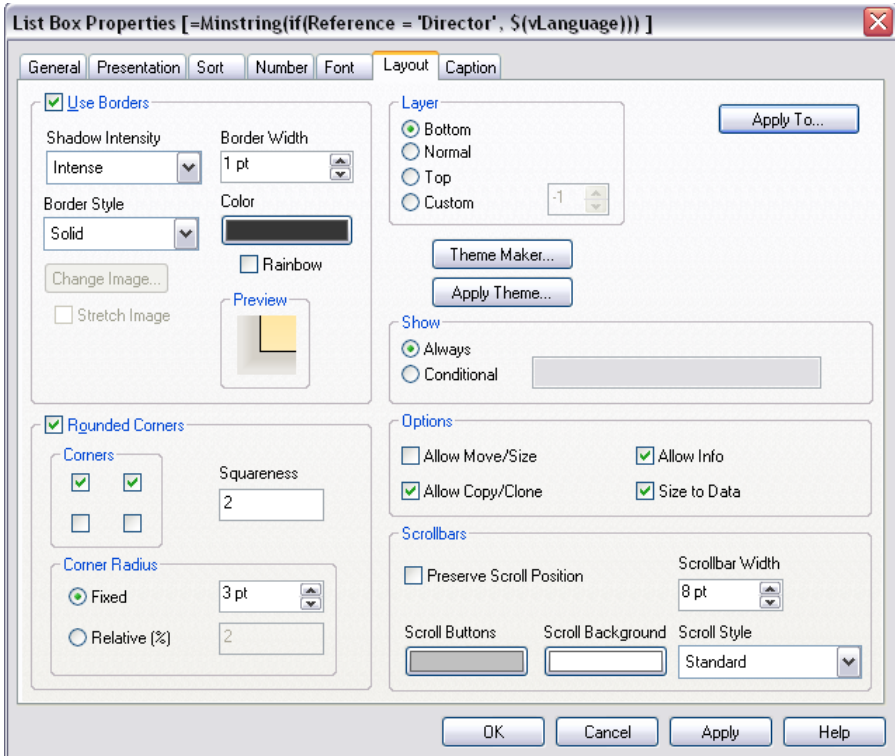


Figure 34. The Layout page in the List Box Properties dialog

In the **Layout** page, you can specify how the list box should appear on the layout. This includes settings for shape, border, and the layer on which it should reside.

The **Layout** page is common to all sheet objects, although the use of some settings is restricted to certain types of sheet objects. The list box is therefore referred to as the sheet object below.

Use Borders

Mark this check box in order to use a border around the sheet object. You specify the type of border by selecting in the drop-down menu. The **Shadow Intensity** drop-down menu makes it possible to set the intensity of the shadow that surrounds the sheet objects. There is also the choice of **No Shadow**.

Border Width

Here you can set the width of the border. Available for all border types. When setting the width of image borders, make sure the width corresponds to the corner width defined in the image. The width can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Border Style

There are the following predefined border types:

Solid

A solid unicolored border.

Depressed

Border giving the impression of depressing the sheet object from the background.

Raised

Border giving the impression of raising the sheet object from the background.

Walled

Border giving the impression of a wall around the sheet object.

Image

Border specified by custom image.

Change Image

If the **Image** border type is selected, you can click this button to set the image you wish to use.

Stretch Image

If the **Image** border type is used and this check box is marked the image pattern will be stretched to fit to the entire space between the corners. If it is not marked, as many copies as possible of the pattern will be displayed, tiled.

Most of the border types can be specified further with the help of the following settings:

Color

Click this button to open a dialog in which you can choose an appropriate base color from the color palette for all border types except **Image**.

Rainbow

Creates a rainbow colored border for all border types except **Image**. The rainbow will start with the selected base color on top of the sheet object.

When **Simplified** is the chosen **Styling Mode**, there is no choice of border type, there is only the **Shadow Intensity** drop-down menu and the **Border Width** setting.

For an example of creating an image border see page 63.

Rounded Corners

In the **Rounded Corners** group, settings can be made for the general shape of the sheet object. Mark this check box to use rounded shapes for the corners of the sheet object. It is possible to achieve shapes from rectangular via super elliptic to elliptic/circular. When **Simplified** is the chosen **Styling Mode**, the **rounded corners** setting is unavailable.

Corners

By marking each of these four check boxes the rounding options are turned on for the respective corners of the sheet object. Corners for which the check box remain unmarked will be drawn rectangular.

Squareness

A number between 2 and 100 setting the general shape of the rounded corners. The number 2 corresponds to a perfect ellipse (or circle for a 1:1 aspect ratio) whereas higher numbers move over super elliptic shapes towards a rectangle.

Corner Radius

This setting determines the radius of the corners in fixed distance or as a percent of the total quadrant. This setting lets you control the extent to which the corners will be affected by the underlying general shape set under **Squareness**. A fixed distance can be specified in mm, cm, inches ("), inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Layer

In the **Layer** group, the sheet object can be assigned one of three layers on the sheet.

Bottom

A sheet object with the **Bottom** layer property can never obscure sheet objects in the **Normal** and **Top** layers. It can

only be placed on top of other sheet objects in the **Bottom** layer.

Normal

When created, sheet objects reside in the normal (middle) layer. A sheet object in the **Normal** layer can never be obscured by sheet objects in the **Bottom** layer and can never obscure sheet objects in the **Top** layer.

Top

A sheet object in the **Top** layer can never be obscured by sheet objects in the **Normal** and **Bottom** layers. Only other sheet objects in the **Top** layer can be placed on top of it.

Custom

The **Top**, **Normal** and **Bottom** layers correspond to internally numbered layers 1, 0 and -1 respectively. In fact all values between -128 and 127 are accepted. Choose this option to enter a value of your choice.

Theme Maker...

Invokes the **Theme Maker Wizard** for creation of a new theme or editing an existing theme. See page 275 for details on QlikView layout themes.

Apply Theme...

Opens a file browser dialog for opening a QlikView layout theme file to be applied to the sheet object. See “Layout Themes” on page 275.

Show

In the **Show** group, it is possible to specify a condition under which the sheet object is shown.

Always

The sheet object will always be shown.

Conditional

The sheet object will be shown or hidden depending on a condition expression which will be evaluated each time the sheet object is to be drawn. The sheet object will only be visible when the condition returns true.

Users with Admin privileges for the document can override all show conditions with the **Show All Sheets and Sheet Objects** in the **Security** page of **Document Properties** (see page 42). This functionality can be toggled by pressing CTRL+SHIFT+S.

Options

In the **Options** group, it is possible to disallow moving/resizing of the sheet object.

Allow Move/Size

If this check box is deselected, it will be impossible for the user to move or resize the sheet object.

Allow Copy/Clone

If this check box is deselected it will become impossible for the user to copy or clone the sheet object.

Allow Info

By default, an info icon **i** will appear in the top right corner of the sheet object if information is linked to the selected field value. Deselect this check box if you do not wish the icon to be displayed. This option is available only for list boxes and multi boxes.

Size to Data

Normally the borders around all table sheet objects in QlikView will shrink when selections cause the size of the table to be less than allocated size for the sheet object. By deselecting this check box this automatic adjustment of size will be turned off leaving any surplus space blank.

Scroll bars

In this group you can set the appearance of scroll bars in the QlikView sheet object.

Preserve Scroll Position

When marking this check box and the corresponding box in **User Preferences, Objects** page, QlikView will try to preserve the scroll position after a selection in table objects and charts with an x-axis scroll bar.

Scroll Bar Background

Sets the scroll bar background color. Select color by clicking the button. Note that medium gray tones often render the best results for scroll bars.

Scroll Buttons

Sets the scroll button color. Select color by clicking the button. Note that medium gray tones often render the best results for scroll bars.

Scroll Bar Width

Sets the scroll bar width. The width can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Scroll Style

Sets the scroll bar style. Choose between the **Scroll Styles Classic, Standard** and **Light**. Scroll Style is not available when **Simplified** is the chosen **Styling Mode**.

Apply to...

Opens the **Caption and Border Properties** dialog (see below), where you can set where to apply the properties you set on the **Layout** page.

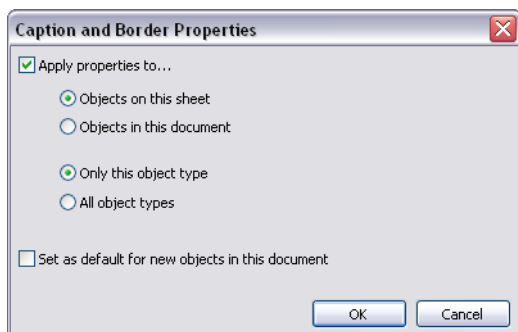
Caption and Border Properties

Figure 35. The Caption and Border properties page

In this dialog you set in where in the document your caption and border properties should apply.

Apply Properties to...

Mark this check box to apply your settings to other than the current object.

Mark one of the following radio buttons.

Objects on this sheet

Applies your settings to objects on the current sheet only.

Objects in this document

Applies your settings to objects of the entire document.

Mark one of the following radio buttons:

Only this object type

Applies your settings to all objects of this type.

All object types

Applies your settings to all objects.

Set as default for new objects in this document

Mark this check box to use your settings as default for all new objects in the current document.

Caption

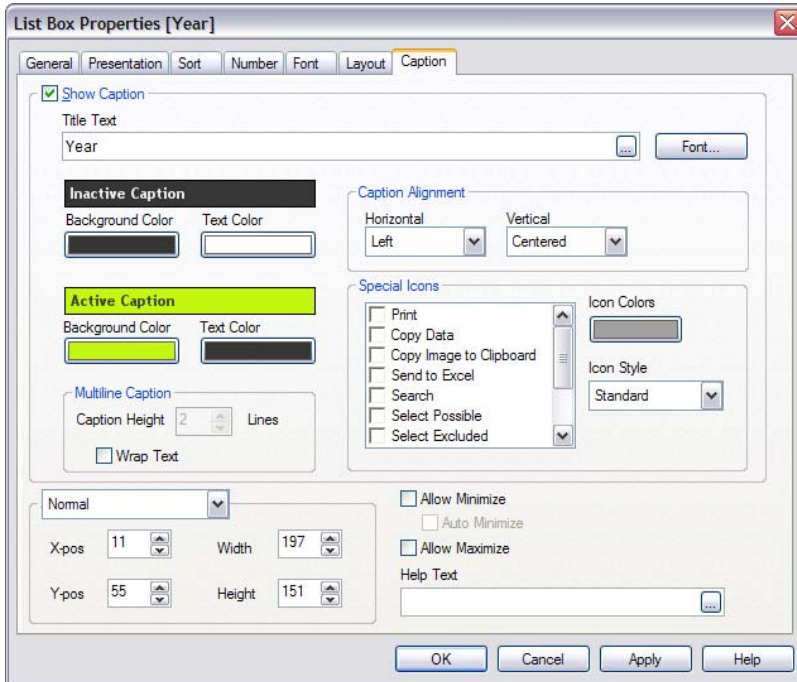


Figure 36. The Caption page in the List Box Properties dialog

In the **Caption** page, advanced settings can be made for the caption. Background and foreground text color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction.

Show Caption

If this check box is marked, a caption will be drawn at the top of the sheet object. List boxes and other "box objects" will have caption turned on by default, whereas buttons, text areas and line/arrow objects will have it turned off.

Title Text

The text to be displayed in the sheet object caption. This setting is identical to the **Window Title** setting found in the **General** page of the properties dialogs of a number of sheet objects. However, for objects such as the text objects, this is the only place where the caption text can be edited. The **Font...** button opens the **Font Dialog** where the font of the caption text can be set.

Inactive Caption / Active Caption

In this group you can set the colors of the caption in its different states. A preview pane shows the current settings of the selected state.

Background Color

Sets the background color. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the color button.

Text Color

Click the colored button to select a color.

Multiline Caption

In this group you can specify multiple line captions.

Caption Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the caption should have.

Wrap Text

If this check box is marked, the caption will be wrapped in two or more lines.

Normal/Minimized Size/Position

In this group you can set exact pixel size and position of the QlikView sheet object in its normal or minimized state. Select state in the drop-down control.

X-pos

Sets the horizontal position of the left side of the sheet object in relation to the sheet's left edge. The setting is made in pixels.

Y-pos

Sets the vertical position of the top side of the sheet object in relation to the sheet's upper edge. The setting is made in pixels.

Width

Sets the width of the sheet object in pixels.

Height

Sets the height of the sheet object in pixels.

Caption Alignment

In this group you set text alignment for caption text.

Horizontal

Sets the horizontal alignment of the caption text. The options are **Left**, **Center** and **Right**.

Vertical

Sets the vertical alignment of the caption text. The options are **Top**, **Center** and **Bottom**. This setting is only relevant when **Multiline Caption** is used.

Special Icons

Many of the object menu commands of the sheet objects can be configured as caption icons. Select commands to be shown as caption icons by marking the check box to the left of each command in the list.

Icon Colors

Sets the background color of the caption icons. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the color button. This setting applies to all icons in the sheet object, not only the special caption icons.

Icon Style

Sets the icon style. Select a style in the drop-down control. This setting applies to all icons in the sheet object, not only the special caption icons.

Note Use the special caption icons with care. One or two of them may provide great help for the end-user while too many may create a “Christmas tree” that severely degrades usability.

Allow Minimize

If this check box is marked, it is possible to turn the sheet object into an icon. The minimize icon will appear in the sheet object caption.

Auto Minimize


This setting, which makes it possible to toggle between sheet objects, should be applied to several sheet objects (preferably positioned on top of each other) on one and the same sheet. Only one of the sheet objects will have its full

size at any given time: as soon as you restore a minimized sheet object with auto minimize set, all other sheet objects with auto minimize set will turn into icons. This option is designed mainly for charts and tables, but may be applied to all sheet objects except buttons, text objects and line/arrow objects.

Allow Maximize

If this check box is marked, it is possible to enlarge the sheet object to fill the sheet. The maximize icon will appear in the sheet object caption.

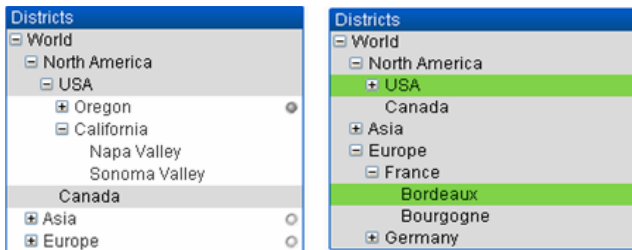
Help Text

Opens a text box in which you can type an optional help message. The text will be displayed when you position the cursor on the help icon  in the top right corner of the sheet object. The text may be a calculated formula (see page 467).

4.4 List Box Tree View

If a field contains paths of nodes in a hierarchy, e.g. ‘World/North America/USA/California/NapaValley’, a list box can be displayed as a tree view. The tree view will show the nodes with indentation to facilitate navigation and allow the user to collapse nodes so that one row in the list box represents a branch in the hierarchy, rather than an individual node.

Selections will be made in the usual manner and multiple selections across different levels in the hierarchy are of course supported.



List box tree view

For expanded nodes and leaves (bottom nodes) the QlikView standard color coding is used. Collapsed nodes, however, may contain a mix of different states. In such a case, the node is displayed green only if all values have been selected, gray if all values are excluded, and white in other cases. In addition, a small beacon is displayed, indicating the status of the hidden nodes.

5 THE STATISTICS BOX

NetSales	
Sum	\$1,565,525
Average	\$721
Min	\$2
Max	\$32,075

Figure 37. The Statistics box

For some fields, it is rarely interesting to look at the individual field values, e.g. the invoice amount in a list of invoices. The sum or average could be very interesting to look at, though. For such fields it is better to use a statistics box than a list box.

Statistics boxes can show most types of statistical entities such as sum, average, minimum etc. of the possible values of a field. The calculations are made dynamically, i.e. the display changes instantaneously as selections are made in sheet objects. The name of the statistics box is by default the same as that of the field to which the selected values belong. When no specific range of values is selected, QlikView treats all the possible values listed in the corresponding sheet object as the statistical sample.

For fields used for association between tables, many of the statistical entities will be unavailable (n/a). See page 463 in Book I for more detailed information.

Selections can also be made by clicking some of the statistical quantities, e.g. Min, Max and Median. The selection is then marked in the sheet object, and not in the statistics box.

5.1 Creation

Statistics boxes can be created by choosing **Create Statistics Box** from the statistics box **Object** menu or by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu. Another possibility is to click the **Create Statistics Box** button in the toolbar.



5.2 Statistics Box Object Menu

The statistics box **Object** menu is found as the **Object** menu when a statistics box is active. It can also be opened as a context menu by clicking with the right mouse button on a statistics box. The menu contains the following commands:

Properties...

Opens the **Statistics Box Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Select Possible

Selects possible field values of the statistics box.

Select Excluded

Selects the currently excluded field values of the statistics box.

Select All

Selects all the field values of the statistics box.

Clear

Deselects all the values of the statistics box. This command can also be invoked via the keyboard shortcut CTRL+D.

Clear Other Fields

Selects all possible values in the field shown in the current statistics box, then clears the selections in all the other fields.

Lock

Locks the selected values of the statistics box. This command can also be invoked via the keyboard shortcut CTRL+L.

Unlock

Unlocks the locked values of the statistics box. This command can also be invoked via the keyboard shortcut CTRL+U.

Print...

Opens the standard **Print** dialog box (see page 263), allowing you to print the contents of the statistics box.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports the statistics box to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer.

Export...

Opens a dialog where you can export the statistics box as a table to a file of your choice. The file formats offered include a range of delimited text file formats: HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files, see page 497 in Book 1).

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Data

Copies the data rows of the statistics box to the clipboard.

Cell Value

Copies the text value of the statistics box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the current statistics box from the sheet.

5.3 Statistics Box Properties

The **Statistics Box Properties** dialog box (Figure 38) is opened by selecting **Properties** from the statistics box **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Statistics Box Properties** dialog box contains several pages:

General

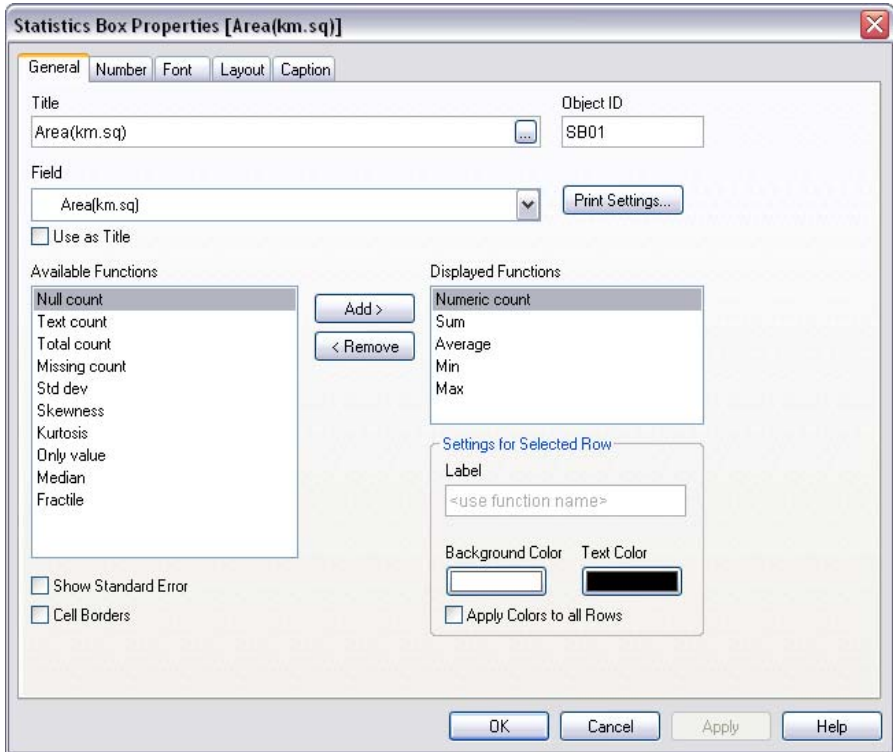


Figure 38. The General page in the Statistics Box Properties dialog

On this page it is possible to configure what statistical entities to show in the statistics box. When a statistics box is opened, *Total count*, *Sum*, *Average*, *Min* and *Max* are displayed by default. There is, however, a number of additional statistics that you can choose from.

The term sample used below is the *statistical sample*, i.e. the sample of possible, numeric values of a field. All numbers refer to the number of records in the first loaded table that contains the field in question.

Title

An alternative name for the statistics box can be entered here. It will be used as window title for the sheet object. The title may be defined as a calculated label expression (see page 467) for dynamic update. If no title is stated, the name of the field shown in the statistics box will be used. This is indicated in the edit box as <use field name>.

Field

The field to be shown in the statistics box.

Use as Title

By selecting this check box, the field name will be used as **Title**. The text <use field name> will appear in the **Title** edit box.

Object ID

The unique ID of the current statistics box. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first statistics box of a document will be assigned the ID SB01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Available Functions

Available statistical functions are listed in this column.

Displayed Functions

Displayed statistical functions are listed in this column.

Add >

Moves the selected statistical functions from the column of available statistical functions to the column of displayed statistical functions.

< Remove

Moves the selected statistical functions listed in the column of displayed statistical functions to the column of available statistical functions.

Show Standard Error

Select this check box to display the standard error of the average and the standard deviation.

Cell Borders

Select this check box to get border lines between the cells displayed in the statistics box.

Settings for Selected Row

In this group individual settings can be made for the row selected in the **Displayed Functions** list above.

Label

An alternative name for the selected statistical function can be entered here.

Background Color

Specifies the background color of the selected row. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Text Color

Specifies the text color of the selected row.

Apply Colors to All Rows

By selecting this check box before clicking **Apply** or **OK** the selected colors will be applied to all rows of the statistics box.

Select the statistical functions to be displayed/removed by clicking on them. Use the **Add >** or **< Remove** buttons to move them to the desired column.

The following statistical functions are available:

Numeric count

Sample size. (Number of numeric values among the possible values.)

Null count

Number of empty field values among the possible values.

Text count

Number of alphanumeric values among the possible values.

Total count

Total number of possible values. This is the same number as the frequency that can be shown in a list box. (Sum of the *Numeric count* and *Text count* described above.)

Missing count

Number of non-numeric values among the possible values. (Sum of *Null count* and *Text count*.)

Sum

Sample sum.

Average

Sample arithmetic mean (average value).

Std dev

Sample standard deviation.

Skewness

Sample skewness.

Kurtosis

Sample kurtosis.

Min

Sample minimum.

Max

Sample maximum.

Only value

Only possible numeric value.

Median

Sample median.

Fractile

Sample fractile.

Number

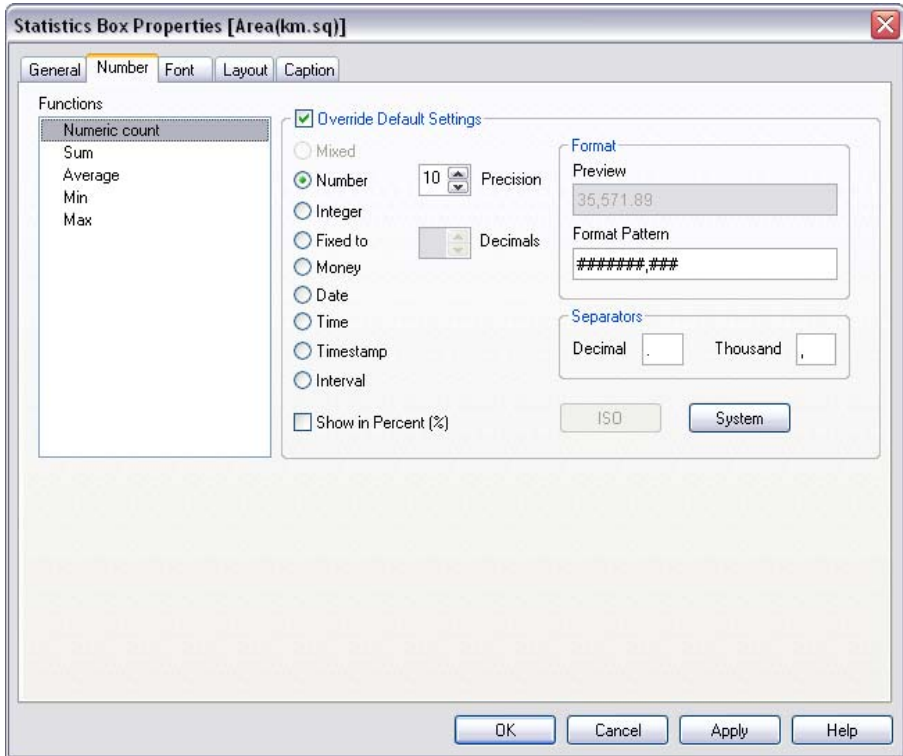


Figure 39. The Number page in the Statistics Box Properties dialog

Statistically correct number formats for the various statistical entities in a statistics box are automatically derived from the base field's data and number format. In the Number page, it is possible to set override number formats for the individual rows of the statistics box, if you disagree with these defaults.

Functions

A list showing all statistical functions currently shown in the statistics box. It is possible to select a function for which the date, time or number format is to be changed.

Override Default Settings

In order to use a custom number format for a specific row, mark this check box and specify a number format section (see page 58).

Font

In the **Font** page (Figure 33 on page 113), you can change the font of the text in the statistics box.

You can specify the font, style and size of the text. The font chosen applies to all the text contained in the statistics box.

Layout

On the **Layout** page (Figure 34 on page 114) you can specify how the statistics box should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120) advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.

6 THE CHART

Charts and tables are sheet objects that can show numbers very compactly. You can e.g. show sums of money, distributed over different fields such as year, month, account number, etc. Numbers that are calculated using several records in the input tables (sums, averages, min, max) can only be shown in charts or statistics boxes.

Charts and tables can be set to display either the frequencies of different values of a field, or a calculated entity, e.g. the sum of the possible values of a field. In both cases, a certain field must be chosen as x-axis, i.e. this field will be used to label the slices of the pie, the different bars in the bar chart and the rows in the pivot table, respectively.

6.1 QlikView Charts

QlikView charts can be divided into two major categories. The first, graph charts, consists of bar, line, combo, pie, scatter, radar, grid, block and gauge charts. The second category, table charts, consists of straight tables and pivot tables. These chart types are drawn as tables with cells in columns and rows. Please note that table boxes, although in many respects similar to table charts are not charts but a separate type of sheet objects. See Book III for more information about charts.



7 THE MULTI BOX



Figure 40. The multi box

The multi box is a sheet object that shows several fields simultaneously.

For each field in the multi box, you will also find a selection indicator: a small beacon that will give you information on the content of the multi box. A green selection indicator shows that there are selected values, a white that there are optional values, and a gray that there are no possible values in the drop-down list.

7.1 Creation

Multi boxes are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu or by pressing the **Create Multi Box** button in the toolbar.



7.2 Multi Box Object Menu

The multi box **Object** menu is found as the **Object** menu when a multi box is active. It can also be opened as a context menu by clicking with the right mouse button on a multi box. The number of available commands differs depending on whether you right-click on the header of the multi box or on one of the fields. In the first case, the field-specific commands are not shown.

Properties...

Opens the **Multi Box Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Select Possible

Selects the possible field values of the field on which you click.

Select Excluded

Selects the currently excluded field values of the field on which you click.

Select All

Selects all the field values of the field on which you click.

Clear

Deselects all the values of the field on which you click. This command can also be invoked via the keyboard shortcut CTRL+D.

Clear Other Fields

Selects all possible values in the current field, then clears the selections in all the other fields.

Lock

Locks the selected values of the field on which you click. This command can also be invoked via the keyboard shortcut CTRL+L.

Unlock

Unlocks the locked values of the field on which you click. This command can also be invoked via the keyboard shortcut CTRL+U.

Clear All Selections

Deselects the values of all the fields in the multi box.

Lock All Selections

Locks the values of all the fields in the multi box.

Unlock All Selections

Unlocks the values of all the fields in the multi box.

Print...

Opens the standard **Print** dialog box (see page 263), allowing you to print the multi box. Use CTRL+P to open the dialog with the keyboard shortcut.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports the multi box to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer.

Export...

Opens a dialog where you can export the multi box as a table to a file of your choice. The file formats offered include a range of delimited text file formats: HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files, see page in Book 1).

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Data

Copies the data rows of the multi box to the clipboard.

Cell Value

Copies the text value of the multi box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.

Object


Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.

Minimize


Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimiz-

ing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the current multi box from the sheet.

7.3 Multi Box Properties

The **Multi Box Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably do not have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

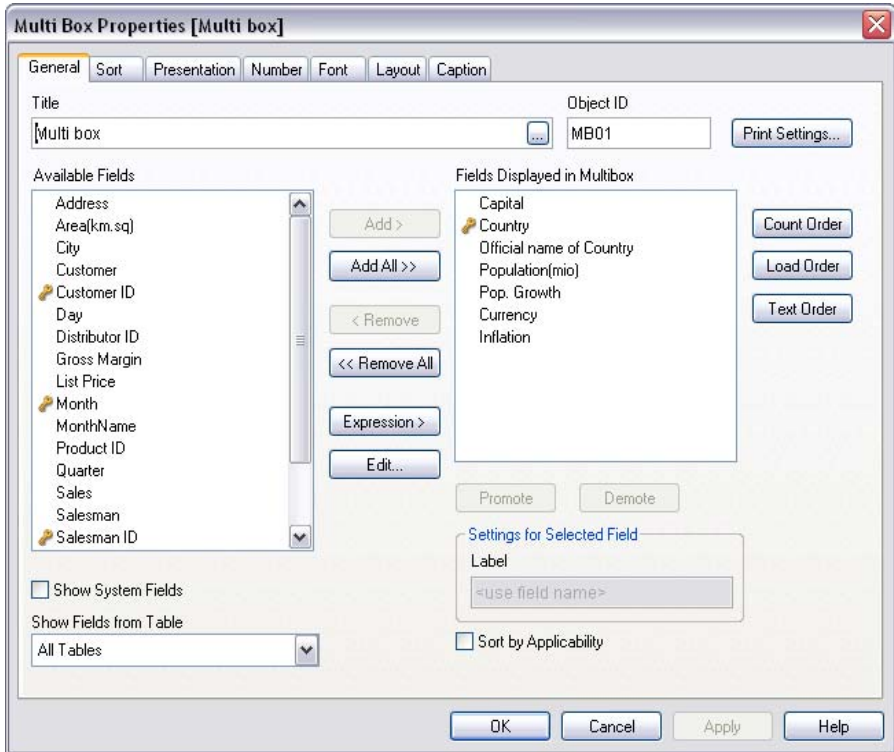


Figure 41. The General page in the Multi Box Properties dialog

The **General** page contains some general settings for the multi box:

Title

Sets the title for the multi box. This name will be shown as window header. The title may be defined as a calculated formula (see page 467) for dynamic update.

Available Fields

The available fields (including the system fields, see above) are listed in this column.

Show System Fields

Check this box to make the system fields appear in the **Available Fields** column.

Show Fields from Table

Normally this drop-down shows the value **-All Tables-** and the **Available Fields/Groups** list contains all fields and groups of the

document. By selecting a specific table name in the drop-down list you may limit the **Available Fields/Groups** list to only fields from that input table.

Finally you may select - **All Tables (Qualified)** - which will show all fields in the document qualified with their table name. A field will appear one time for each table it appears in.

Fields Displayed in Multi box

Displayed fields are listed in this column. Initially this column is empty.

Add >

Moves the selected fields from the column of available fields to the column of displayed fields.

Add All >>

Moves all the fields in the column of available fields to the column of displayed fields.

< Remove

Moves the selected fields listed in the column of displayed fields to the column of available fields.

<< Remove All

Moves all the fields from the column of displayed fields to the column of available fields.

Expression

Opens the **Edit Expression** dialog. Create an expression to be used as a displayed field in the Multi Box.

Edit...

Opens the **Edit Expression** dialog for the field chosen in the **Fields Displayed in Multibox**.

Select the fields to be used/removed by clicking on them. Use the **Add >** or **< Remove** buttons to move them to the desired column.

Count Order >>

Sorts the fields in the column of displayed fields according to the number of field values in each field.

Load Order >>

Sorts the fields in the column of displayed fields according to the load order of the fields.

Text Order >>

Sorts the fields in the column of displayed fields according to the alphabetical order of the fields.

Sort by Applicability

Sorts the fields in the multi box according to whether they contain possible field values or not. Fields lacking possible field values are dynamically moved down if this check box is marked.

Note The sort commands on this page affect the sort order of the fields, not the field values. The sort order for field values is set on the **Sort** page.

Promote

Moves the selected field one step up.

Demote

Moves the selected field one step down.

Object ID

The unique ID of the current multi box. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first multi box of a document will be assigned the ID MB01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Label

An alternative name for the displayed field can be entered here. It will be used as field title in the multi box. The label may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Sort

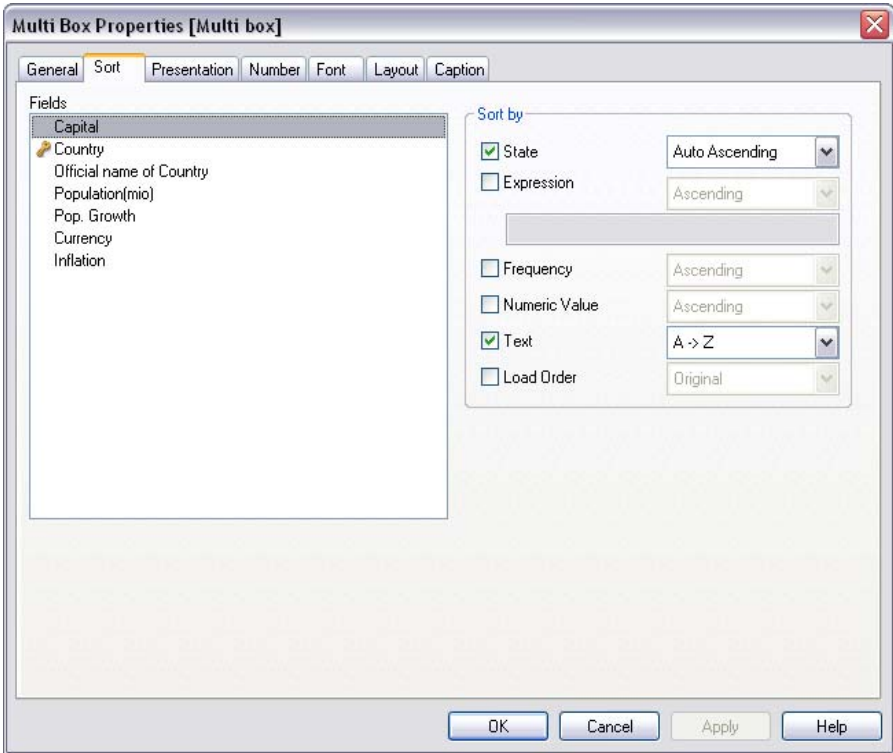


Figure 42. The Sort page in the Multi Box Properties dialog

In the **Sort** page the sort order of the field values can be set for each field in the multi box.

Fields

In this group it is possible to select one or several fields for which the sort order is to be set.

Sort by

In this group, the sort order of the selected field's values can be set.

State

Sorts the field values according to their logical state (selected, optional or excluded).

If **Auto Ascending** is set and the multi box is large enough to display all values, the box will be sorted according to the second sort order set, **Ascending** if no other sort order is

defined. If not all values can be displayed in the multi box, the sort order will be **Ascending**.

Expression

Sorts the field values according to the expression entered into the text edit box below this sort option. The label may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Frequency

Sorts the field values by frequency (number of occurrences in the table).

Numeric Value

Sorts the field values by their numeric values.

Text

Sorts the field values in alphabetical order.

Load order

Sorts the field values by the initial load order.

The order of priority is **State, Expression, Frequency, Numeric Value, Text** and **Load Order**. Each of these sort criteria can be set to ascending or descending. Note that the sort commands on this sheet affect the sort order of the field values, not of the fields themselves. The sort order for fields are set on the **General** page.

Presentation

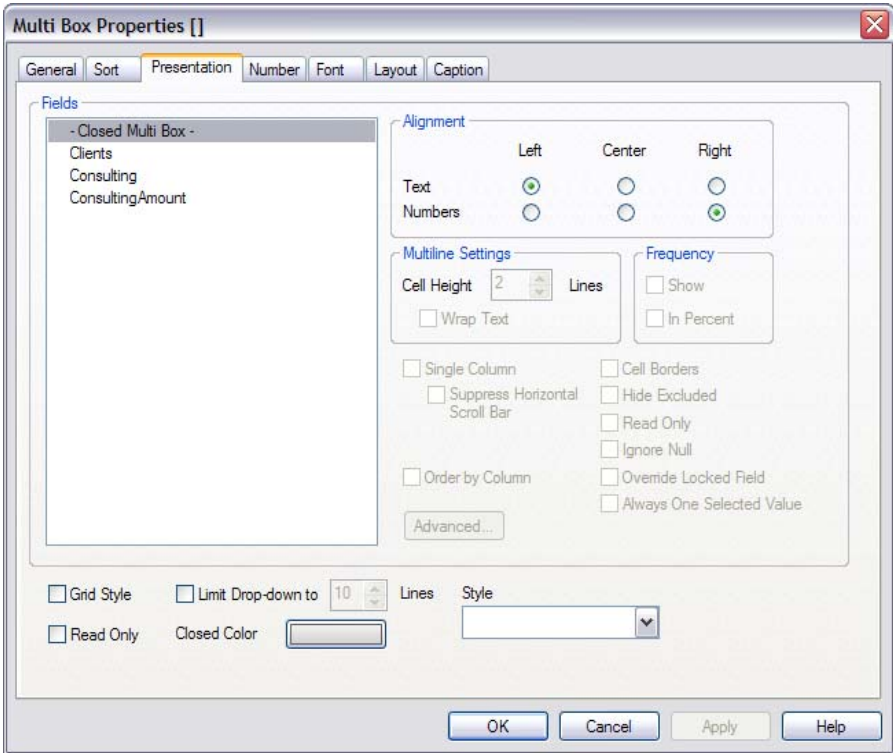


Figure 43. The Presentation page in the Multi Box Properties dialog

On the **Presentation** page it is possible to set properties concerning the presentation of data in the multi box.

Fields

In this group you can select one or more fields for which you wish to change the presentation settings. In addition to the fields found in the multi box, the list contains the item ' - closed multi box - '. Select this item to change the settings of the closed multi box.

Alignment

In this group, the alignment of field values in the selected fields can be set. **Text** and **Numbers** can be individually set to left-adjusted, centered or right-adjusted.

Multiline Settings

In this group you can set the default display of multiple line cells for the selected fields.

Wrap Text

If this check box is marked, the contents of the cell will be wrapped in two or more lines.

Cell Height *n* Lines

If you have selected **Wrap Text**, this is where you set how many lines the cell should have.

Frequency

In the **Frequency** group you specify if and how frequency of a field value should be shown in selected fields.

Show

Toggles the status as to whether or not the frequency of a field value is shown in selected fields. Frequency is here the number of records containing possible values of a field. This option is normally not available for fields used for associations between tables (see page 463 in Book I).

In Percent

Toggles the status as to whether or not the frequency should be shown in absolute numbers or as percentages of the total number of entries.

Single Column

Marking this check box will force single column mode in the selected fields. If it is not checked, you will automatically get several columns within the field value list.

Suppress Horizontal Scroll Bar

Mark this check box to suppress horizontal scroll bar in list box when field values are too wide to be displayed in their entirety.

Order by Column

If this check box is marked, a multiple column presentation will be ordered by columns instead of rows.

Cell Borders

Mark this check box to get horizontal border lines between the different field values in the selected fields.

Hide Excluded

Toggles whether or not the excluded values of the selected fields should be shown.

Read Only

Marking this check box makes selections in the multi box impossible for the selected fields.

Ignore NULL

The value in a specific row of the multi box will only be shown if there is exactly one possible value to show. When QlikView counts possible values it will also count NULL values. If you want NULL values to be excluded in the count mark this check box. Incorrect use of this option may lead to the display of non-correlated data in the multi box.

Override Locked Field

If this check box is marked, selections can be made in the field from the multi box even if the field is locked. The field will still be locked for logical changes stemming from selections in other fields.

Always One Selected Value

If this check box is marked, the multi box will always have one and only one value selected. This is useful when creating applications where one value always has to be selected in order for charts etc. to make sense. An example could be the selection of one currency rate. When this option is marked, it is not possible to select an excluded value in another field in the multi box. The option is available only when you have already selected exactly one value in the multi box. This option is not available for calculated fields.

Advanced...

Opens the **Advanced Field Settings** dialog (see page 149) which offers settings for image representation of field values and special text search options.

Grid Style



OrderID	<input type="checkbox"/>	Customer
OrderDate	<input type="checkbox"/>	Country
SalesPerson	<input type="checkbox"/>	Shipper

Marking the **Grid Style** check box changes the appearance of the multi box (see the picture above).

Read Only

Marking this check box makes selections in the multi box impossible.

Limit Drop-down to n Lines

By marking this check box you can limit the length of opened drop-down list boxes in the multi box. Enter the maximum number of values to be shown in the edit box.

Closed Color

Specifies the color of the data column cells of a closed multi box. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Style

Choose between the styles **Borders** and **Light**.

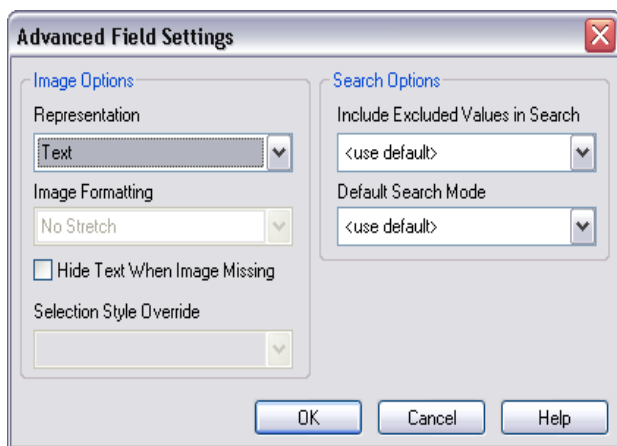
Advanced Field Settings dialog

Figure 44. Advanced Field Settings

This dialog contains settings for image representation of field values and special text search options when using fields for text searches in drop-downs. Open this dialog from the **Presentation** pages in **Multi Box Properties**, **Table Box Properties** or **Chart Properties**, **Dimensions** page.

Image Options

This group allows you to have a field value interpreted as a reference to an image in memory or on disk. The following settings are possible:

Representation

Three alternatives are given

Text

When selecting this option the field values will always be interpreted and displayed as text.

Image

When selecting this option QlikView will try to interpret each field value as a reference to an image. The reference may be a path to an image file on disk (e.g. `C:\MyPic.jpg`) or inside the qvw document (e.g. `qmem://<Name>/<Peter>`). If QlikView cannot interpret a field value as a valid image reference, the value itself will be displayed.

Info as Image

When selecting this option QlikView will display image info linked to the field value via **info load/select** in the script. If no image info is available for a field value, the value itself will be displayed. Option not available for bitmap charts.

Image Formatting

Only available when one of the image options has been selected above. This setting describes how QlikView formats the image to fit in the cell. There are four alternatives.

No Stretch

If this option is selected, the image will be shown as is, without any stretching. This may cause parts of the picture to be invisible or only part of the cell to be filled.

Fill

If this option is selected, the image will be stretched to fit the cell without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the image will be stretched as far as possible to fill the cell while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the image will be stretched to fill the cell in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Hide Text When Image Missing

If this option is selected, QlikView will not display the field value text if interpretation as an image reference fails for some reason.

Selection Style Override

When images are shown instead of text it may be necessary to use another selection style than the document default so that the logical state of the field values becomes visible. Use the drop-down to select a suitable selection style, e.g. Corner Tags. This option is only available when image representation of some form has been selected under **Representation**.

Search Options

This group allows you to control certain aspects of text search applicable to opened multi boxes and drop-down select in tables.

Include Excluded Values in Search

Specifies whether excluded values should be included in text searches. Three alternatives are given.

<use default>

The default as specified under **User Preferences** will apply. (See page 95 in Book I.)

Yes

Excluded values are always included in text search.

No

Excluded values are not included in text search. Only optional values are included.

Default Search Mode

Specifies the default search mode to be used in text search. The mode can always be intermittently changed by typing * or ~ as part of the search string. Three alternatives are given:

<use default>

The default as specified under **User Preferences** will apply. (See page 95 in Book I.)

Use Normal Search

Normal search is default when starting a text search.

Use Fuzzy Search

Fuzzy search is default when starting a text search.

Spacing

Mark this check box to insert extra empty spacing in tables wherever the value of the selected dimension changes. The setting can be made in the edit box in increments of 1/16 of current line height. The empty lines logically belong to the line above, to the effect that clicking on them will cause selection to appear in the preceding value. This check box is only available for table charts.

Number

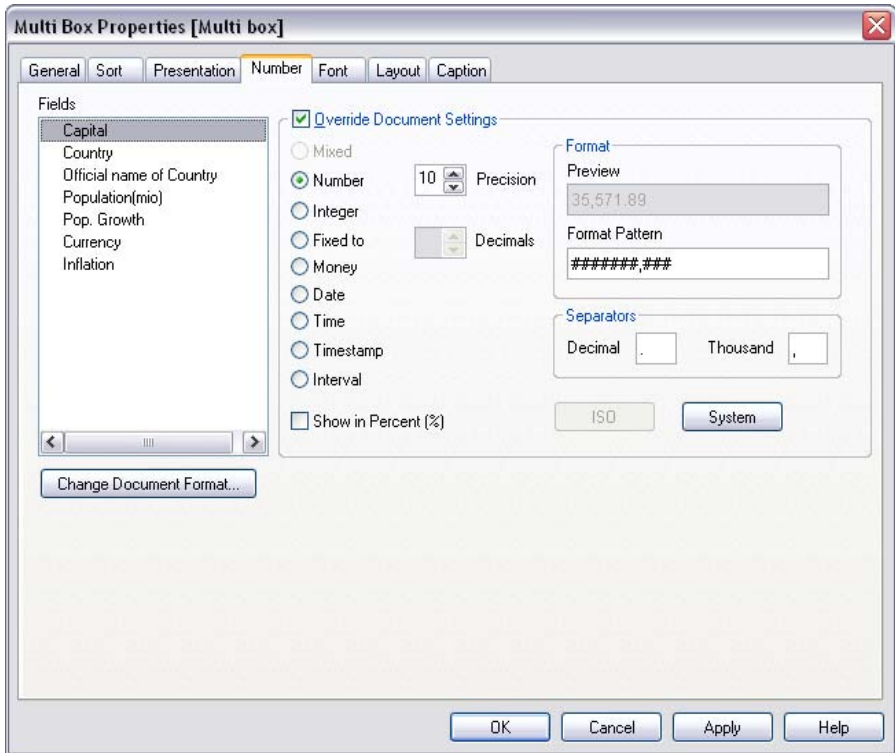


Figure 45. The Number page of the Multi box properties dialog

In the **Number** page, it is possible to set the number format for the different fields of the multi box.

Fields

A list showing all fields currently shown in the multi box. It is possible to select a field for which the date, time or number format is to be changed.

Override Document Settings

Each field has a number format which can be set on the **Number** page of the **Document Properties** dialog. Normally this is the number format used when displaying field values in a multi box. To do this, mark this check box and specify a number format in the group control below. For an explanation of the controls, see page 301 in Book II.

Change Document Format...

Opens the **Number** page of the **Document Properties** dialog, where you can change the underlying number format of the fields.

Font

On the **Font** page (Figure 33 on page 113), you can change the font of the text in the multi box.

You can specify the font, style and size of the text. The font chosen applies to all text contained in the multi box.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the multi box should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.



8 THE TABLE BOX

Suppliers		
Supplier	Category	Product
ABC	Baby Clothes	Mehmet-Napp
ABC	Baby Clothes	Mehmet-Skor
ABC	Baby Clothes	Mehmet-Tröja
Asin Fashion Ltd Co	Swimwear	Summer Shorts
Asin Fashion Ltd Co	Women's Clothes	Okkaba Skin Jackets
Asin Fashion Ltd Co	Women's Footwear	Walking Shoes
Austerlich	Children's Clothes	RDL Suit
Austerlich	Men's Clothes	Bow tie
Austerlich	Men's Footwear	Davenport Shoes

Figure 46. The table box

The table box is a sheet object that shows several fields simultaneously. The content is record-oriented in the same way as a normal table, i.e. the contents of one row are logically connected. The columns of the table box can be fetched from different input tables, which gives the user a possibility to create a new table with all the logically possible combinations of the values in the fields of the input tables.

8.1 Creation

Table boxes are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu, or by pressing the **Create Table Box** button in the toolbar.



8.2 Using the Table Box

Sorting

It is possible to sort the table box by any column: simply right-click on the column and choose **Sort** from the context menu. You can also double-click the caption of column you want to sort by. Choosing **Sort** from the context menu or double-clicking the same column caption will invert the sort order.

Changing the Column Order

The order of the columns can be changed by means of the drag-and-drop-method. Point at the column title, then keep the mouse button depressed while dragging the column to its new position. You can disable this feature by deselecting the **Allow Drag and Drop** check box in the **Table Box Properties** dialog, **Presentation** page.

8.3 Table Box Object Menu

The table box **Object** menu is found as the **Object** menu when a table box is active. It can also be opened as a context menu by clicking with the right mouse button on a table box. The number of available commands differs depending on whether you right-click on the header of the table box or on one of the columns. In the first case, field specific commands are not shown.

Properties...

Opens the **Table Box Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Fit Columns to Data

Adjusts the width of all columns in the tables to the widest data in each column. The header is included in the calculation.

Equal Columnwidth

Sets the columnwidth in the table equal to that of the column on which you clicked.

Sort

Sorts the records by the field on which you click.

Custom Format Cell

Opens the **Custom Format Cell** dialog (see page 163) which lets you format cells in the column you clicked upon. The command is only available when design grid is turned on or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I).

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Change Value

Only available for table box columns displaying input fields. Sets the cell that has been clicked in input edit mode. Equivalent to clicking the input icon in the cell.

Restore Values

Only available for table boxes displaying input fields. Opens a menu with the following options.

Restore Single Value

Restores the value in the field value that has been clicked to its default value from the script.

Restore Possible Values

Restores the values of all possible field values to their default values from the script.

Restore All Values

Restores the values of all field values to their default values from the script.

Select Possible

Selects the possible field values of the field on which you click.

Select Excluded

Selects the currently excluded field values of the field on which you click.

Select All

Selects all the field values of the field on which you click.

Clear All Selections

Clears selections in all fields displayed in the table box.

Clear

Deselects all the values of the field on which you click.

Clear Other Fields

Selects all possible values in the current field, then clears the selections in all the other fields.

Lock

Locks the selected values of the field on which you click.

Unlock

Unlocks the locked values of the field on which you click.

Print...

Opens the standard **Print** dialog box (see page 263), allowing you to print the table. Use CTRL+P to open this dialog with the keyboard shortcut.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports the table to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet. For this functionality to work, Microsoft Excel 97 or later must be installed on the computer.

Export...

Exports the contents of the table to a file of your choice. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data files, see page 497 in Book I).

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Full Table

Copies the table to the clipboard, complete with header and selection status.

Table Data Area

Copies only the values of the table to the clipboard.

Cell Value

Copies the text value of the table box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the current table box from the sheet.

8.4 Custom Format Cell Dialog

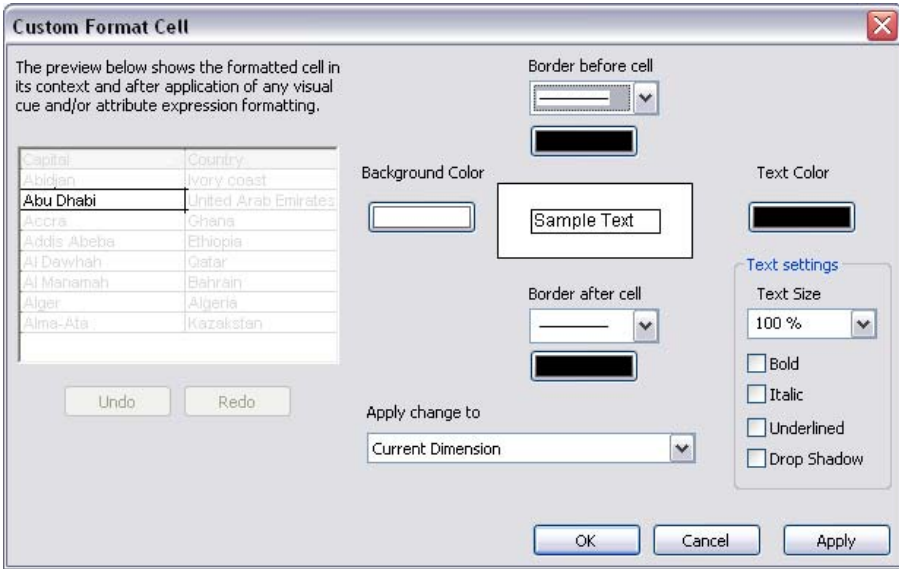


Figure 47. The Custom Format Cell dialog

This dialog lets you apply custom formats to one or more groups of table cells. It can be invoked from the object menus of a table box, a straight table or a pivot table, provided that the design grid is turned on or **Always Show Design Menu Items** is marked in **User Preferences** (see page 106 in Book I).

The group of cells to be affected by the formatting is determined by where you right-click in the table to invoke the command. The smallest group of cells to be formatted as a unit is one field (normally column) in a table box or one expression/dimension in a table chart. If stripping is used, each stripe is formatted separately.

Whenever this dialog is used to custom format any part of a table, the **Style** as set in the **Style** page of the table's Properties dialog will be set to **[Custom]**. If you change the **Style** setting back to one of the pre-defined styles, the custom formatting will be lost.

Note Table styles in general will be superseded by any formatting resulting from attribute expressions.

Table Preview Pane

The preview pane shows format changes made to the group/groups of table cells. Unlike the smaller preview to the right, **Sample Text**,

it shows the changes in context and after the application of visual cues and/or attribute expression formatting.

Within this pane it is possible to freely move about the table and apply formats to the different groups of cells. Just click in a cell to move formatting focus.

Undo

The **Undo** button can be used for undoing the changes made within the **Custom Format Cell** dialog step by step.

Redo

The **Redo** button can be used for reapplying changes previously undone with the **Undo** button within the **Custom Format Cell** dialog step by step.

After leaving the **Custom Format Cell** dialog all changes made in the dialog may be undone with just one click on the **Undo** button in the main toolbar.

Sample Text

This is the preview cell which shows the settings made.

Background Color

Defines the cell background color.

Text Color

Defines the cell text color.

Border Before Cell

Here you define the border to precede the cell. There is a drop-down for setting border style and a button to set border color. Note that cell borders will be merged between cells, which may cause the actual border in the table to look differently from the preview.

Border After Cell

Here you define the border to succeed the cell. There is a drop-down for setting border style and a button to set border color. Note that cell borders will be merged between cells, which may cause the actual border in the table to look differently from the preview.

Text Settings

In this group you find a number of text modifiers applicable to the cell.

Size

Use the drop-down to apply a size modification to the general font used in the table.

Bold

Mark this check box for bold text.

Italic

Mark this check box for italic text.

Underline

Mark this check box for underlined text.

Drop Shadow

Mark this check box for text with drop shadow.

Apply Change To

Changes are normally applied only to the expression, dimension or field (table boxes) in which you right-clicked to access the dialog.

With the help of this drop-down you may choose to apply the same formatting to other expressions, dimensions or fields.

8.5 Table Box Properties

The **Table Box Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

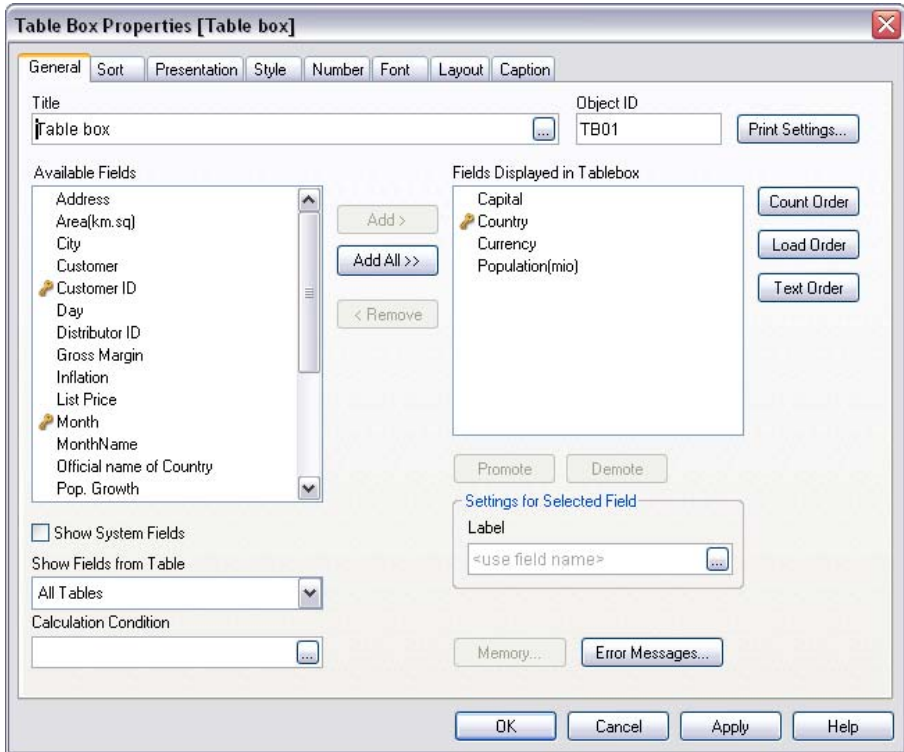


Figure 48. The General page in the Table Box Properties dialog

The **General** page contains some general settings for the Table box.

Title

Sets the title for the table box. This name will be shown as window header. The title may be defined as a calculated formula (see page 467) for dynamic update. By clicking the **...** button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Available Fields

Available fields (including the system fields, see below) are listed in this column. Provided that the document script has been executed all fields which appear in more than one input table will be preceded with a key symbol.

Show Systems Fields

Check this box to get the system fields visible in the **Available Fields** column.

Show Fields from Table

Normally this drop-down shows the value **-All Tables-** and the **Available Fields/Groups** list contains all fields and groups of the document. By selecting a specific table name in the drop-down list you may limit the **Available Fields/Groups** list to only fields from that input table.

Finally you may select **-All Tables (Qualified)** - which will show all fields in the document qualified with their table name. A field will appear one time for each table it appears in.

Calculation Condition

Typing an expression in this text box sets a condition to be fulfilled for the table box to be calculated. If the condition is not fulfilled, the text 'Calculation condition unfulfilled' will be displayed in the table box.

Fields Displayed in Table box

Displayed fields (columns in the table box) are listed in this column. Initially this column is empty.

Add >

Moves the selected fields from the column of available fields to the column of displayed fields.

Add All >>

Moves all the fields in the column of available fields to the column of displayed fields.

< Remove

Moves the selected fields listed in the column of displayed fields to the column of available fields.

Select the fields to be used/removed by clicking on them. Use the **Add >** or **< Remove** buttons to move them to the desired column.

Count Order

Sorts the fields in the column of available fields according to the number of field values in each field. If you click twice, the order will be reversed.

Load Order

Sorts the fields in the column of displayed fields according to the load order of the fields. If you click twice, the order will be reversed.

Text Order

Sorts the fields in the column of displayed fields according to the alphabetical order of the fields. If you click twice, the order will be reversed.

Note that the sort commands on this sheet affect the sort order of the fields, i.e. the column order, not the order of the records. The sort order for field values is set on the **Sort** page.

Promote

Moves the selected field one step to the left.

Demote

Moves the selected field one step to the right.

Object ID

The unique ID of the current table box. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first table box of a document will be assigned the ID TB01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Label

An alternative name for the displayed field can be entered here. It will be used as column title in the table box. The label may be defined as a calculated formula (see page 467) for dynamic update. By clicking the **...** button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Error Messages

Opens a dialog in which you can define custom error messages to be displayed instead of the standard ones. See further on page 471.

Memory...

Opens the **Calculation Memory Settings** dialog (see further on page 125 in Book I), in which it is possible to increase the memory allocated to the table. This might be necessary for large tables. This button is not available when **Ignore Calculation Limits** has been selected in the **Objects** page of the **User Preferences** dialog.

Sort

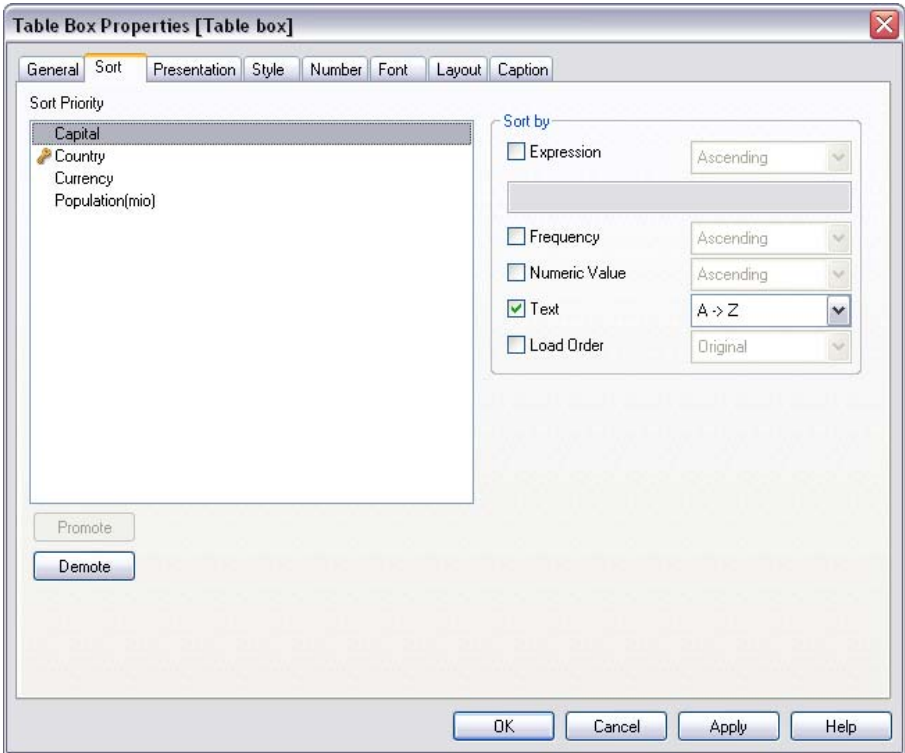


Figure 49. The Sort page in the Table Box Properties dialog

In the **Sort** page the sort order of the field values and of the records can be set. The order defined here is used also when the **Sort** command in the **Object** menu is chosen, or when you double-click in a column header.

Sort Priority

In this group it is possible to select one or several fields for which the sort order is to be set. The order of the fields in this column defines the order of priority when sorting the records.

Promote

Moves the selected field one step up in the **Fields** list.

Demote

Moves the selected field one step down in the **Fields** list.

Sort by

In this group the sort order of the selected fields can be set.

Expression

Sorts the field values according to the expression entered into the text edit box below this sort option.

Frequency

Sorts the field values by frequency (number of occurrences in the table).

Numeric Value

Sorts the field values by their numeric values.

Text

Sorts the field values in alphabetical order.

Load Order

Sorts the field values by the initial load order.

The order of priority within a field is **Expression, Frequency, Numeric Value, Text** and **Load Order**. Each of these sort criteria can be set to ascending or descending. The order of priority between the fields is set by the order of the fields in the **Fields** list.

Note that the sort commands on this sheet affect the sort order of the field values, not the columns. The sort order for columns are set on the **General** page.

Presentation

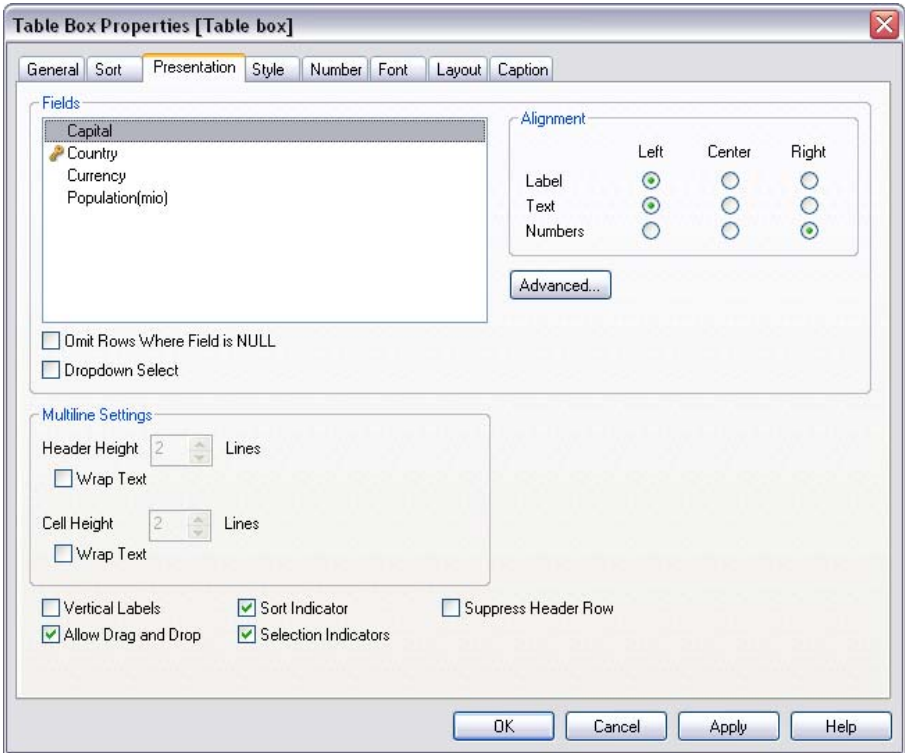


Figure 50. The Presentation page in the Table box Properties dialog

In the **Presentation** page it is possible to set properties concerning the presentation of data in the table box.

Fields

In this group it is possible to select one or several fields for which the presentation properties are to be set.

Alignment

In the **Alignment** group, the alignment of field values of the selected fields can be set. **Label**, **Text** and **Numbers** can be individually set to left-adjusted, centered or right-adjusted.

Omit Rows where Field is NULL

Marking this check box will force the table to omit records in which the field value is missing. If it is not checked, you will see missing field values as gray cells.

Drop-down Select

If this check box is selected for a field column, a drop-down icon will appear to the left in the column header. By clicking the icon a list box showing all field values of the field will be opened over the table. You may then make selections and searches in the same manner as if the field had been a row in a multi box.

Advanced

Opens the **Advanced Field Settings** dialog (see page 149) which offers settings for image representation of field values and special text search options.

Multiline Settings

In this group you can set the height of the table header and data row cells.

Header Height (Lines)

If you have select **Wrap Text**, this is where you set how many lines the header cells should have.

Wrap Text

If this check box is marked, the contents of the header row cells will be wrapped in two or more lines.

Cell Height (Lines)

If you select **Wrap Text**, this is where you set how many lines the data cells should have.

Wrap Text

If this check box is marked, the contents of the data cells will be wrapped in two or more lines.

Vertical Labels

If this check box is marked, the column titles are displayed vertically.

Allow Drag and Drop

Deselect this check box to disable drag and drop moves within the column.

Sort Indicator

If this check box is selected, a sort indicator icon (arrow) will be shown in the header of the column, which is at the top of the current column sort order. The direction of the icon indicates whether the column is sorted ascending or descending.

Selection Indicators

If this check box is selected, selection indicators (beacons) will be shown in table columns containing fields with selections. For new

tables this setting takes its default from the **Selection Indicators in Tables** on the **Objects** page of **User Preferences**.

Suppress Header Row

If this check box is marked the table will be shown without header (label) row.

Style

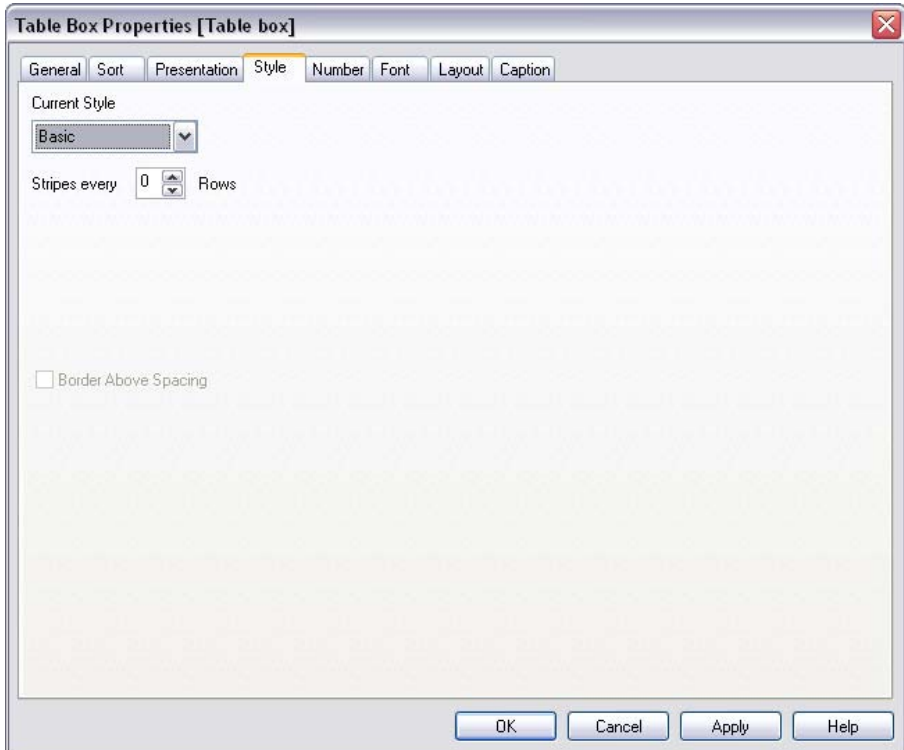


Figure 51. The Style page in the Table Box Properties dialog.

In the **Style** page, you make settings for the table formatting style.

Current Style

You can choose an appropriate table style from the drop-down list. If the value [Custom] appears in the drop-down control a custom style has been applied to the table. If you change the setting back to one of the pre-defined styles, the custom formatting will be lost.

Stripes every n Rows

Here you can specify if and at how long intervals shaded stripes should appear.

Number

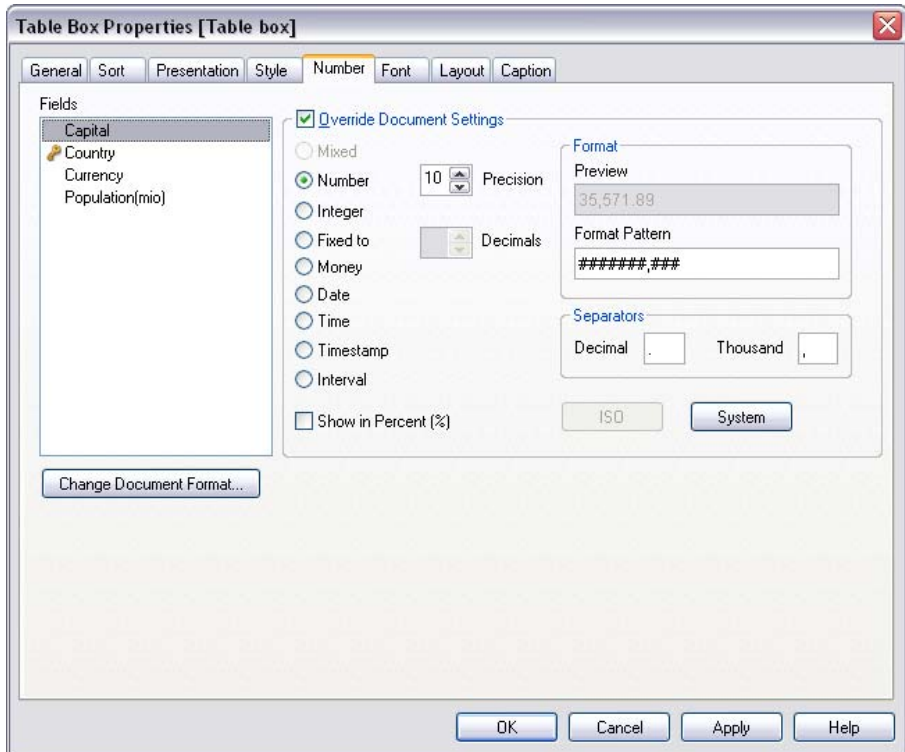


Figure 52. The Number page in the Table Box Properties dialog

In the **Number** page, it is possible to set the number format for the different fields of the table box.

Fields

A list showing all fields currently shown in the table box. It is possible to select a field for which the date, time or number format is to be changed.

Override Document Settings

Each field has a number format which can be set on the **Number** page of the **Document Properties** dialog. Normally this is the number used when displaying field values in a table box. It is however

possible to use a separate number format in an individual table box. To do this, mark this check box and specify a number format in the group control below. For an explanation of the controls, see the number format section, page 301 in Book II.

Change Document Format

Opens the **Number** page of the **Document Properties** dialog, where you can change the underlying number format of the fields.

Font

In the **Font** page (Figure 33 on page 113), you can change the font of the text in the table box.

You can specify the font and size of the text. The font chosen applies to all text contained in the table box. The style is chosen in the **Style** page (see above).

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the table box should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.

9 THE CURRENT SELECTIONS BOX

Current Selections	
Fields	Values
City	Boise
SupplierCo...	Asin Fashion Ltd Co, ABC

Figure 53. The Current Selections box

The current selections box is a sheet object that shows selections in the fields and their logical status. The current selections box shows the same data as the free floating **Current Selections Window** (see page 141 in Book I) but can be positioned on the sheet as a sheet object.

9.1 Creation

Current selection boxes are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu, or by pressing the **Create Current Selections Box** button in the toolbar.



9.2 Current Selections Box Object Menu

The current selections box **Object** menu is found as the **Object** menu when a current selections box is active. It can also be opened as a context menu by clicking with the right mouse button on a current selections box. The number of available commands differs depending on whether you right-click on the header of the current selections box or on one of the fields. In the first case, the field-specific commands are not shown. In the second case commands relating to all fields are not shown. The menu contains the following commands:

Properties...

Opens the **Current Selections Box Properties** dialog. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Select Possible

Selects the possible field values of the field on which you click. This command is only available when you right-click on a specific field.

Select Excluded

Selects the currently excluded field values of the field on which you click. This command is only available when you right-click on a specific field.

Select All

Selects all the field values of the field on which you click. This command is only available when you right-click on a specific field.

Clear

Deselects all the values of the field on which you click. This command is only available when you right-click on a specific field.

Clear Other Fields

Selects all possible values in the current field, then clears the selections in all the other fields. This command is only available when you right-click on a specific field.

Lock

Locks the selected values of the field on which you click. This command is only available when you right-click on a specific field.

Unlock

Unlocks the locked values of the field on which you click. This command is only available when you right-click on a specific field.

Clear All Selections

Clears the selections in all fields in the document. This command is not available when you right-click on a specific field.

Lock All Selections

Locks the selections in all fields in the document. This command is not available when you right-click on a specific field.

Unlock All Selections

Unlocks the selections in all fields in the document. This command is not available when you right-click on a specific field.

Print...

Opens the standard **Print** dialog box (see page 263), allowing you to print the current selections box. Use CTRL+P to open this dialog with the keyboard shortcut.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports the text to Microsoft Excel, which is automatically launched if not already running. The text will appear in the cells of a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer.

Export...

Opens a dialog where you can export the contents in the current selections to a file of your choice. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files, see page 497 in Book I).

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Data

Copies the data (selections) in the selected current selections box to the clipboard.

Cell Value

Copies the text value of the current selections box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.

Image


Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and

border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.



Minimize

Iconizes the current selections box. Clicking  in the object caption (if shown) gives the same result. To restore the object to its previous size and location, double-click the minimized object. This command is available only if you have selected the option **Allow Minimize** on the **Caption** page(see Figure 36 on page 120).

Restore

Restores a minimized or maximized current selections box to its previous size and location. It is also possible to restore a minimized object by double-clicking the caption or clicking , provided that the caption is shown. This command is available only if you have selected at least one of the options **Allow Minimize** or **Allow Maximize** on the **Caption** page (see Figure 36 on page 120) and if the object is in the minimized or maximized state.

Maximize

Enlarges the current selections box to fill the sheet. Clicking  in the object caption (if shown) gives the same result. To restore the list box to its previous size and location, choose **Restore** or click . This command is available only if you have selected the option **Allow Maximize** on the **Caption** page (see Figure 36 on page 120).

Help

Opens context-specific help.

Remove

Removes the active current selections box from the sheet.

9.3 Current Selections Box Properties

The **Current Selections Box Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

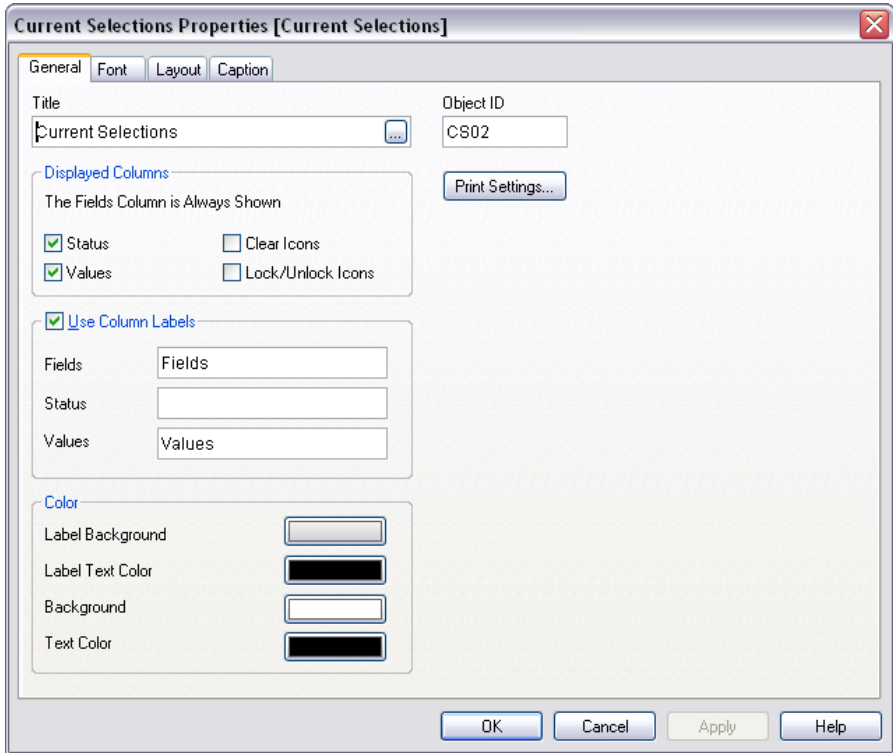


Figure 54. The General page in the Current Selections Box Properties dialog

In the **General** page settings for title and configuration of the Current Selections box can be made.

Title

Sets the title for the current selections box. This name will be shown as window header. The title may be defined as a calculated formula (see page 467) for dynamic update. The default value is "Current Selections".

Displayed Columns

In this group you decide which columns should appear in the current selections box. The **Fields** column is always shown.

Status

If this check box is marked, the current selections box will contain a **Status** column with colored beacons indicating

the current logical status of the selections in the fields (e.g. locked).

Values

If this check box is marked the current selections box will contain a **Values** column showing the values selected in the fields.

Clear Icons

If this check box is marked each field row in the current selections box will display a small clear icon. Clicking the clear icon will clear selections in the field. No clear icon will be shown for locked fields.

Lock/Unlock Icons

If this check box is marked each field row in the current selections box will display a small lock or unlock icon. Clicking the icon will lock or unlock selections in the fields.

Use Column Labels

If this check box is marked a header row will be shown in the current selections box. Labels can be edited in the group below.

Fields

Here you may specify a label to be shown in the header row above the **Fields** column.

Status

Here you may specify a label to be shown in the header row above the **Status** column.

Values

Here you may specify a label to be shown in the header row above the **Values** column.

Colors

In this group you may set the colors of the current selections box.

Label Background

Defines the background color of the label row.

Label Text Color

Defines the text color of the label row.

Background

Defines the background color of the display area.

Text Color

Defines the text color of the display area.

Object ID

The unique ID of the current current selections object. Upon creation, every layout entity is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of entity, as well as of a number. The first current selections object of a document will be assigned the ID CS01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet, alert, bookmark or report in the document.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Font

On the **Font** page (Figure 33 on page 113), you can change the font of the text in the current selections box.

You can specify the font, style and size of the text. The font chosen applies to all text contained in the current selections box.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the current selections box should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.



10 THE INPUT BOX

The input box is a sheet object that is used to enter data into QlikView variables and to show their values.



Figure 55. The Input Box

10.1 Creation

Input boxes are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu or by pressing the **Create Input Box** button in the toolbar.



10.2 Using the Input Box

An input box consists of three columns in a multi box like layout. The first column shows a list of variables. The second column equal signs '=' and the third column the values of the variables. It can contain one or many variables, each one on a separate row.

Variables in QlikView are named entities containing a single data value, unlike fields which can (and normally do) contain multiple values. Also, whereas fields get their values from **load** and **select** statements in the script, variables get their values from **let** and **set** statements in the script, via Automation calls, or by the help of input boxes in the layout. For more information on variables and fields, see chapter “Variables and Fields” on page 213 in Book I.

Variables can contain numeric or alphanumeric data. If the first character of a variable value is an equal sign '=' QlikView will try to evaluate the value as a formula (QlikView expression) and then display or return the result rather than the actual formula text.

In an input box the current value of the variable is shown. By clicking on a value in the input box, the cell will turn into edit mode, so that a new value can be entered or the old one modified. If the variable contains a formula, that will now be shown rather than its result. The cell in edit mode will normally contain a ... button which opens up a full editor window to facilitate creation of advanced formulas. The function of a variable value cell in an input box could very well be compared to that of a cell in a spreadsheet.

Sometimes the input box variable value cell will contain a drop-down icon, giving fast access to recently used values or predefined values. A variable may have input constraints attached to it, barring input of all values which do not meet certain criteria. In certain cases a variable in an input box may be read-only, in which case it is impossible to enter edit mode.

10.3 Input Box Object Menu

The input box **Object** menu is found as the **Object** menu when an input box is active. It can also be opened as a context menu by clicking with the right mouse button on an input box. The number of available commands differs depending on whether you right-click on the header of the input box or on one of the values. In the first case, the field specific commands are not shown. In the second case, commands relating to all fields are not shown.

Properties...

Opens the input box Properties dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Print...

Opens the standard **Print** dialog box (see page 263), allowing you to print the input box. Use CTRL+P to open this dialog with the keyboard shortcut.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports the input box to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer.

Export...

Opens a dialog where you can export the contents of the input box to a file of your choice. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files, see page 497 in Book I).

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Data

Copies the data rows of the input box to the clipboard.

Cell Value

Copies the text value of the input box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.

Minimize


Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available

only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the current input box from the sheet.

10.4 Input Box Properties

The **Input Box Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

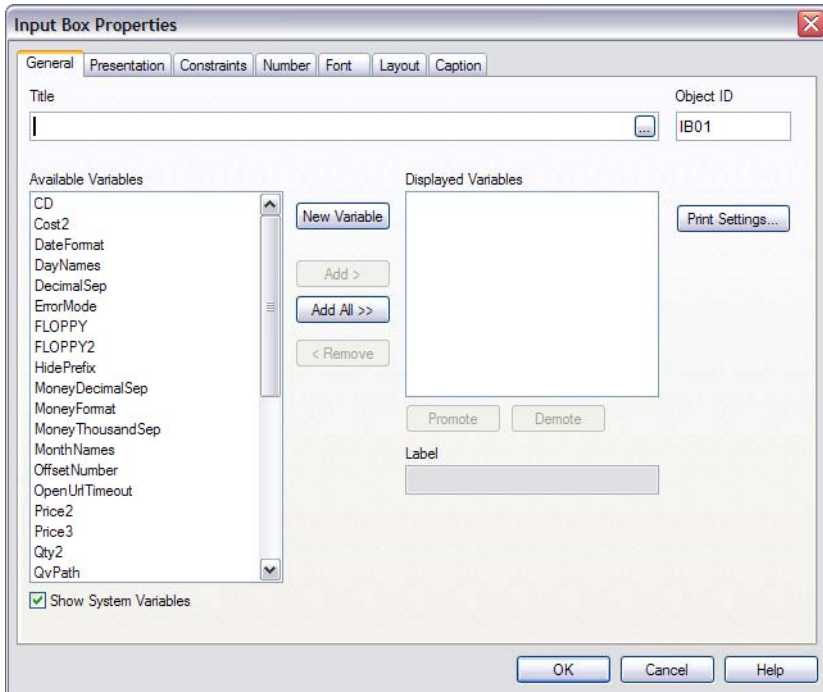


Figure 56. The General Page in the Input Box Properties dialog

In the **General** page you set a title for the input box and select what variables should be displayed.

Title

Sets the title for the input box. This name will be shown as window header. The title may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Available Variables

The available variables (including the system variables, see above) are listed in this column.

Show System Variables

Check this box to make the system variables appear in the **Available Variables** column. System variables are typically created during script execution.

Displayed Variables

Variables to be displayed in the input box are listed in this column. Initially this column is empty.

New Variable

Opens a dialog for creating a new variable which is then placed in the column of displayed variables.

Add

Moves the selected variables from the column of available variables to the column of displayed variables.

Add all>>

Moves all the variables in the column of available variables to the column of displayed variables.

< Remove

Moves the selected variables listed in the column of displayed variables to the column of available variables.

Select the variables to be used/removed by clicking on them. Use the **Add >** or **< Remove** buttons to move them to the desired column.

Promote

Moves the selected variable one step up in the column of displayed variables and thereby also in the order by which they are shown in the input box.

Demote

Moves the selected variable one step down in the column of displayed variables and thereby also in the order by which they are shown in the input box.

Label

An alternative name for the displayed variable can be entered here. It will be used as variable title in the input box. The label may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full Edit Expression dialog is opened for easier editing of long formulas.

Object ID

The unique ID of the current input box. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first input box of a document will be assigned the ID IB01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Presentation

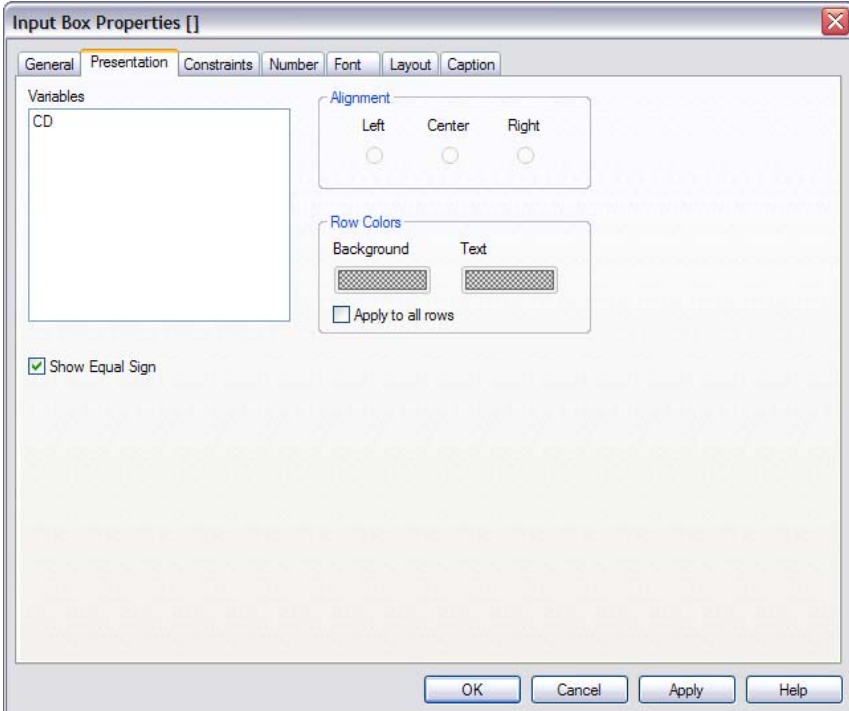


Figure 57. The Presentation page in the properties dialog.

Variables

Shows a list of all variables in the current input box. When a variable name in the list is selected the properties of that variable can be modified.

Show Equal Sign

Uncheck this box if you do not want the equal sign to be displayed in the Input Box. This setting is general for all variables.

Alignment

The alignment of the variables can be set. The variables can be individually set to left-adjusted, centered or right-adjusted.

Row Colors

In this group individual color settings can be made for the row selected in the **Displayed Variables** list to the left.

Background

Specifies the background color of the selected row. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Text

Specifies the text color of the selected row.

Apply to All Rows

By selecting this check box before clicking **Apply** or **OK** the selected colors will be applied to all rows of the input box.

Constraints

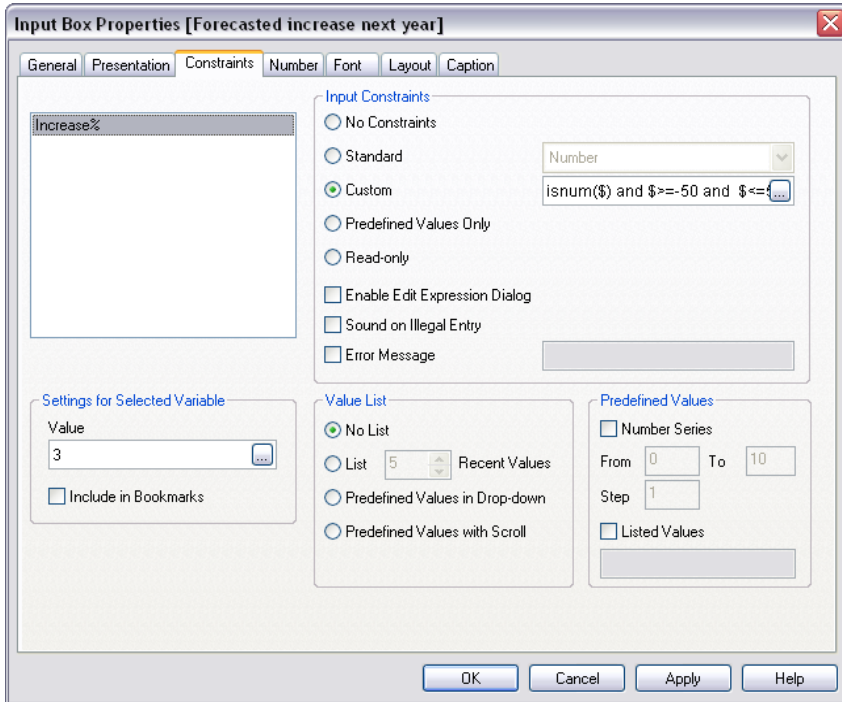


Figure 58. The Constraints page in the *i* Properties dialog

In the **Constraints** page advanced settings can be made for the variables in the input box. For more information on variables, see page 214 in Book I.

Variables

Shows a list of all variables in the current input box. When a variable name in the list is selected the properties of that variable can be modified.

Settings for Selected Variable

In this group you can make settings for a variable selected in the list above.

Value

Shows the current value of the selected variable, which can also be edited.

Include in Bookmarks

Variables are normally not included in bookmarks. Mark this check box if you want the variable to be included in bookmarks.

Input constraints

In the **Input constraints** group you can specify constraints against which all values entered into an input box variable by the user will be checked. If a value does not meet the constraints specified and **Input Constraints** is set to **Predefined Values Only**, it will be rejected and an error message may be displayed (see below).

No Constraints

By default no constraints are selected, i.e. all values can be entered into the variable.

Standard

Input values will be checked against one of a number of common standard constraints which can be selected in the drop-down box.

Custom

Input values will be checked against a user specified constraint. The constraint is entered in the edit box and should be expressed as a QlikView expression returning true (non-zero value) if the input value is acceptable. The input value is referenced as a dollar sign, \$ in the expression (Example : $\$ > 0$ will cause the input box to accept only positive numbers in the selected variable). The previous value of the variable can be referenced by the variable name (Example: $\$ > = abc + 1$ as a constraint on a variable named abc will cause the input box to accept only numeric entries with a value of the old value plus 1).

Predefined Values Only

Input values will be checked against a list of values defined in the **Predefined Values** group (see below). Only input values found in the list will be accepted.

Read-only

Marks the variable as read only. No values can be entered.

Enable Edit Expression Dialog

Mark this check box to enable expansion of the input box edit window to the full **Edit Expression** dialog by means of a ... button.

Sound on Illegal Entry

If this check box is selected, QlikView will issue a sound warning when the user attempts to enter a value which falls outside of the constraints.

Error Message

Normally, if the user attempts to enter a value which falls outside of the constraints it will just be rejected leaving the present variable value in place. By selecting this check box it is possible to specify an error message which will be presented to the user in case of incorrect entries. The error message is typed in the edit box. It may be defined as a calculated formula (see page 467) for dynamic update.

Value List

In the **Value List** group you can specify constraints against which all values entered into an input box variable by the user will be checked. If a value does not meet the constraints specified, it will be rejected and an error message (see above) may be shown.

No List

No list of values previously used in the variable is maintained. No drop-down list with previous or predefined values will be available to the user for the selected variable in the input box.

List N Recent Values

A list of values previously used in the variable is maintained. A drop-down list with the most recently used values will be available to the user for the selected variable in the input box. The number of previous values to store can be set in the edit box

Predefined Values in Drop-Down

A drop-down list with predefined values (see **Predefined Values** below) will be available to the user for the selected variable in the input box.

Predefined Values with Scroll

A scroll control will be available to the user for the selected variable in the input box. The scrolling will be made between the predefined values (see **Predefined Values** below).

Predefined Values

In the **Predefined Values** group you can specify a list of predefined values which can be presented to the user in a drop-down list and/or used to define acceptable variable values.

Number Series

If this check box is selected a list of predefined numeric values will be generated based on a lower limit, an upper limit and a step value. This option can be used alone or in combination with **Listed values** (see below).

From

Defines the lower limit for the number series. Must be a real number.

To

Defines the upper limit for the number series. Must be a real number.

Step

Defines the step between values in the number series. Must be a real number.

Listed Values

By selecting this check box it is possible to specify a list of arbitrary predefined values. The values can be numeric or alphanumeric. Alphanumeric values should be enclosed in quotes (e.g. 'abc'). Each value should be separated by a semicolon; (e.g. 'abc';45;14.3;'xyz'). This option can be used alone or in combination with **Number series** (see above).

Number

On the **Number** page the number format can be set for the variables in the input box. This dialog is very similar to its counterpart under **Document Properties** (see page 58).

Font

On the **Font** page (Figure 33 on page 113), you can change the font of the text in the input box. You can specify the font, style and size of the text. The font chosen applies to all text contained in the input box.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the input box should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.



11 THE BUTTON

Buttons can be used in QlikView to perform commands or actions, e.g. exporting data to files, launching other applications or executing macros.



Clear selections

Figure 59. The Button

11.1 Creation

Buttons are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu or by pressing the **Create Button** button in the toolbar.



11.2 Button Object Menu

The button object **Object** menu is found as the **Object** menu when a button is active. It can also be opened as a context menu by clicking with the right mouse button on a button. The menu contains the following commands:

Properties...

Opens the **Button Properties** dialog box.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Send to Excel

Exports the text to Microsoft Excel, which is automatically launched if not already running. The text will appear in a single cell in a new Excel worksheet.

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Text

Copies the text displayed on the button to the clipboard.

Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.

Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.

Help

Opens context specific help.

Remove

Removes the button from the sheet.

11.3 Button Properties

The **Button Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

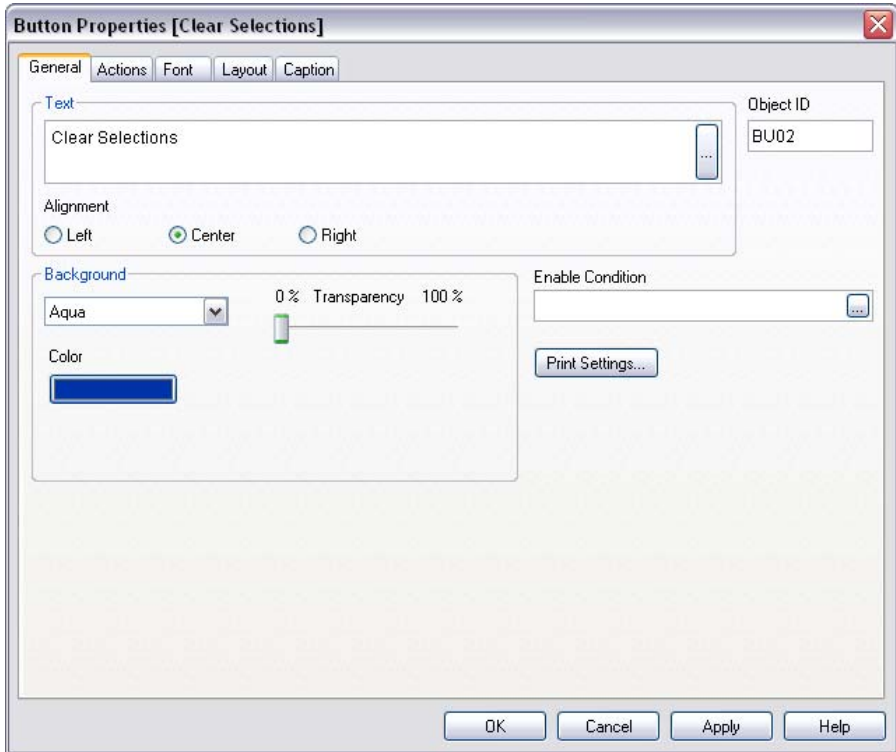


Figure 60. The General page in the Button Properties dialog

In the **Text** group you can set the properties of the text to be displayed on the button.

Text

Marking this control and entering a text in the text field labels the button. The text may be defined as a calculated formula (see page 467 in Book III) for dynamic update. The text may consist of several lines, divided when you press ENTER.

Alignment

The button text can be left-aligned, centered or right-aligned.

Background

In the **Background** group you can specify the background of the button. The drop-down menu gives you the following different basic button styles to choose from.

Aqua

Default for new buttons. Gives a rounded button with three-dimensional glassy look.

Color

The color can be defined as a solid color as a gradient via the **Color Area** dialog that opens when clicking the color button.

Plain

Produces a traditional plain QlikView button.

System Default

Provides a solid background in the color defined for buttons in the operating system.

Color

Select this radio button if you want the button to be displayed with a colored background. The color can be defined as a solid color or as a gradient via the **Color Area** dialog that opens when clicking the color button.

Image

Produces an image button. The image could be either a single static image or a combined image with three parts, one for each of the button's three possible states (active, inactive or depressed).

Combined Image



Mark this radio button and browse to an image file using the **Select Image...** button in order to assign a three-state combined image to the button. The image file must consist of three images, side by side, of the button: the first one of an active button, the second of a depressed button and the third one of a dimmed (inactive) button.

Single Image

Mark this radio button and browse to an image file using the **Select Image...** button in order to assign a single-state image to the button. Image types supported include jpg, png, bmp, gif and animated gif.

Transparency

Sets the degree of transparency of the button background. At 100% the background will be completely transparent. The transparency will apply regardless if a color or an image is used for background.

Object ID

The unique ID of the current button. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first button of an document will be assigned the ID BU01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Enable Condition

The expression entered here determines the status of the button. If the expression returns 0, the button will be disabled; if the expression returns 1, the button will be enabled. If no expression is entered, 1 is assumed. Buttons that are disabled due to underlying status cannot be enabled by means of a condition.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Actions

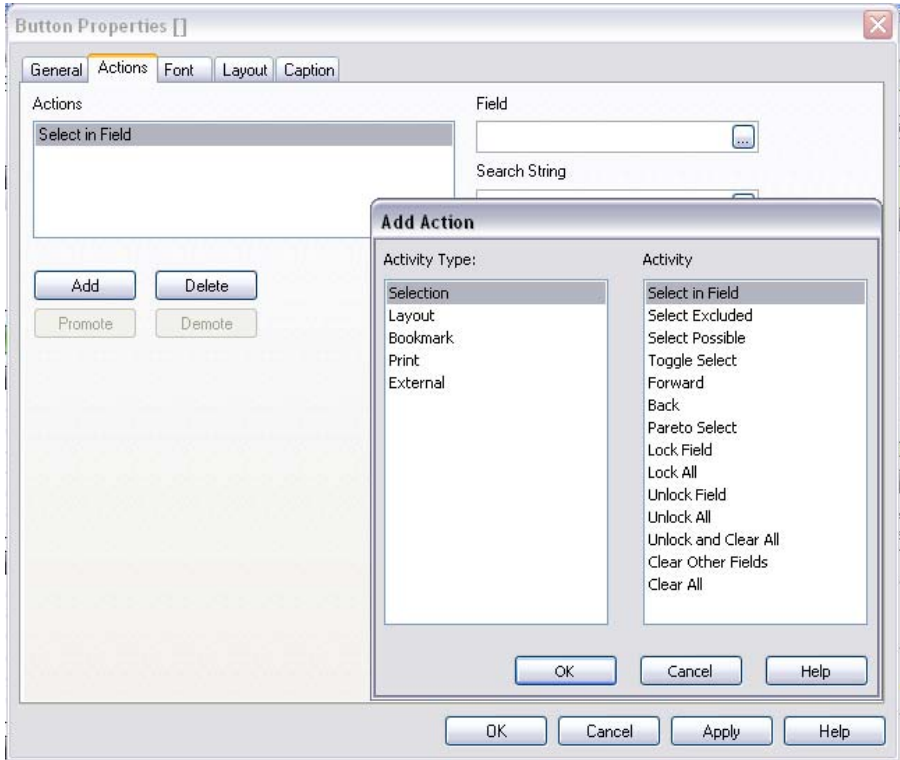


Figure 61. The Actions page in the Button Properties dialog

On this page, the actions of the button are set, for example launching an external application or exporting data from QlikView. In all fields with the ... button it is possible to enter a calculated expression for dynamic update (see page 249 in Book III).

Add

Opens the **Add Action** dialog, see below, where the actions for the button are chosen. Choose the type of action in the list. Depending on the action, different parameters for the action will appear on the **Actions** page. The actions will be executed in the order that they appear in the list.

Delete

Removes the action from the button.

Promote

Moves the selected action up in the list of actions.

Demote

Moves the selected action down in the list of actions.

The following actions can be added from the **Add Action** dialog.

Selection**Select in Field**

Selects the **Field** and **Search String** that are specified.

Select Excluded

Selects the excluded values in the specified **Field**.

Select Possible

Selects the possible values in the specified **Field**.

Toggle Select

Toggles between the current selection and the specified **Field** and **Search String**.

Forward

Goes one step forward in the list of selections.

Back

Goes back one step in the list of selections.

Pareto Select

Makes a pareto selection in the specified **Field** based on an **Expression** and **Percentage**. This type of selection is used to select the top contributors to a measure, typically in line of a general 80/20 rule. For example, to find the top customers that contribute to 80% of the turnover, then *Customer* should be used as field, **sum(Turnover)** should be used as expression and 80 should be used as percentage.

Lock Field

Locks the selections in the specified **Field**.

Lock All

Locks all values in all fields.

Unlock Field

Unlocks the selections in the specified **Field**.

Unlock All

Unlocks all values in all fields.

Unlock and Clear All

Unlocks all values and clears all selections in all field.

Clear Other Fields

Clears all fields except the **Field** specified.

Clear All

Clears all selections except locked ones.

Layout**Activate Object**

Activates the object specified by **Object ID**.

Activate Sheet

Activates the sheet specified by **Sheet ID**.

Activate Next Sheet

Opens the next sheet in the document.

Activate Previous Sheet

Opens the previous sheet in the document.

Minimize Object

Minimizes the object specified by **Object ID**.

Maximize Object

Maximizes the object specified by **Object ID**.

Restore Object

Restores the object specified by **Object ID**.

Bookmark**Apply Bookmark**

Applies the bookmark specified by **Bookmark ID**. If two bookmarks have the same ID, the document bookmark is applied. To apply the server bookmark specify Server\bookmark ID.

Create Bookmark

Creates a bookmark from the current selection. Specify **Bookmark ID** and **Bookmark Name**.

Replace Bookmark

Replaces the bookmark specified by **Bookmark ID** with the current selection.

Print¹**Print Object**

Prints the object specified by **Object ID**. Specify the **Printer Name** if the object should be sent to another printer than the default printer.

Print Sheet

Prints the sheet specified by **Sheet ID**.

1. Not available as document and sheet trigger!

Print Report

Prints the report specified by **Report ID**. Specify the **Printer Name** if the report should be sent to another printer than the default printer.

Mark the check box **Show Print Dialog** if you want the Windows print dialog to be shown.

External**Export¹**

Exports a table containing a specific set of fields, but only those records that are applicable according to the made selection are exported. Click on the **Setup** button on the **Actions** page to open the **Export Action Settings** dialog (see below).

Launch¹

Launches an external program. The following settings can be configured:

Application

Click on **Browse...** to find the application that should be launched.

File Name

Enter the path to the file that should be opened with the application specified above.

Parameters

Specify parameters for the command line from which the application is started.

Working Directory

Sets the working directory for the application to be launched.

Exit application when QlikView is closed

Forces the application to close when QlikView is exited.

Open URL¹

Opens the **URL** in the default web browser.

Open QlikView Document¹

Opens the specified QlikView **Document** and applies the selections of the original document. The file extension must be included. Mark the check box **Transfer State** to transfer

1. Not available as document and sheet trigger!

the selections from the original document to the one you wish to open. The opened document will first be cleared of selections. Mark **Apply state on top of current** to retain the second document's selections and apply the original document's selections on top of them.

Run Macro

Enter the path and name of the macro to be run in **Macro Name**. You can type any name for which you later create a macro in the **Edit Module** dialog, or a calculated expression for dynamic update (see page 249 in Book III). The macro (if it exists) will be executed when the button is clicked. See page 207 below about macros.

Set Variable

Assigns a **Value** to the specified **Variable**.

Show Information¹

Shows the associated information, such as a text file or an image for the field specified by **Field**.

Close This Document¹

Closes the active QlikView document.

Note Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

1. Not available as document and sheet trigger!

The Export Action Settings dialog

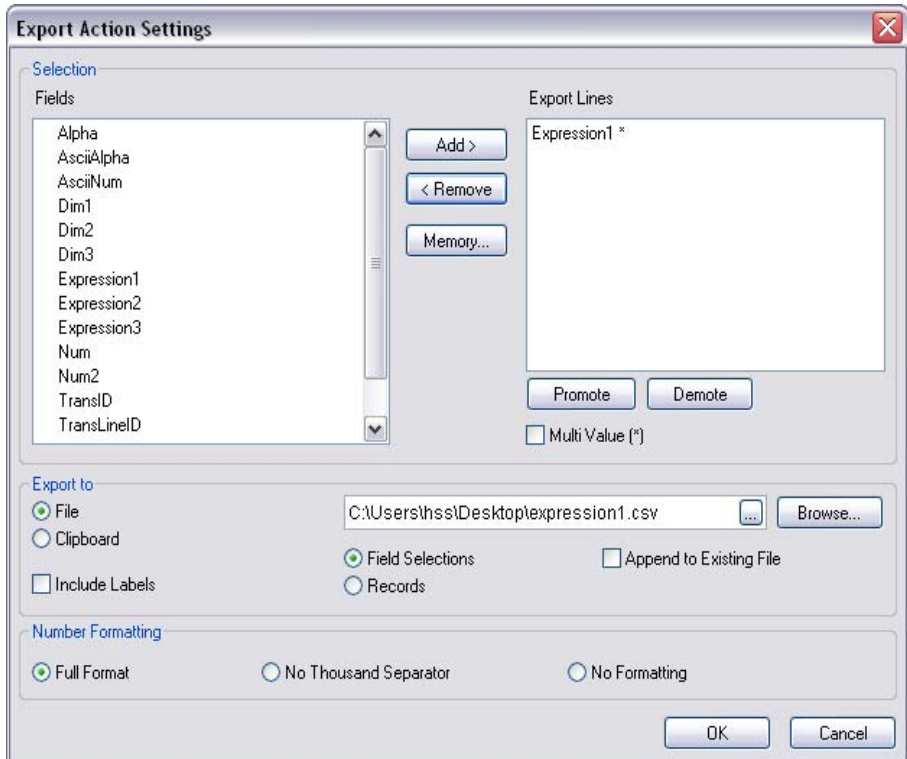


Figure 62. The Export Action Settings page

The **Export Action Settings** dialog contains the following options:

Selection

In this group you set what fields should be selected for export.

Fields

In the **Fields** box you find a list of the available fields.

Export Lines

In this box you find the fields you have selected for export. The fields for which you have allowed multiple values are marked with an asterisk.

Add >

Adds fields to the **Export Lines** box.

< Remove

Removes fields from the **Export Lines** box.

Memory...

Opens the **Calculation Memory Settings** dialog (see page 125 in Book I), where you can increase the allocated amount of memory.

Promote

Moves the selected field one step up, i.e. one step to the left in the export table.

Demote

Moves the selected field one step down, i.e. one step to the right in the export table.

Multi Value (*)

By marking a field in the **Export Lines** box and checking this control, you allow a field to have several values in the export listing.

Export to

In this group you can choose whether you want to export the values to a file or to the clipboard.

File

If you export to a file you must mark this check box and enter the file name. If no file name is entered, the values will be exported to the clipboard. The file name may be entered as a calculated formula.

Browse

Opens the **Export File** dialog (Table 86 on page 274), allowing you to browse for a file to which the values should be exported.

Clipboard

Default setting. If no file is specified above, the values will be exported to the clipboard.

Field Selections

If this control is checked, the export file will, for each selected field, have all the possible field values on one row, separated by tabs.

Include Labels

If this control is checked, the first position on the line (when **Field Selections** is set) or the first record (when **Records** is set) will contain the field names.

Records

If this control is checked, the export file will have one column for each selected field, separated by tabs.

Append to Existing File

If this control is checked the export will be appended to the export file, if the file already exists. Labels will not be exported when appending to an existing file. If the export file does not exist, this flag has no meaning.

Number Formatting

Some other programs may have difficulties to handle numbers with number format correctly. QlikView offers three options for number formatting of numeric data to be exported to files or to the clipboard.

New export buttons will inherit the default setting from the **Export** page in the **User Preferences** dialog. The setting can however be individually set for each export button.

Full Format

Instructs QlikView to export numeric data with its full number format, just as it is shown in the document's sheet objects.

No Thousand Separator

Removes any thousand separator from numeric data. This option is recommended if data are to be imported into MS Excel.

No Formatting

Removes all number formatting from the data and exports the raw numbers. The decimal separator will be as defined in the system settings (**Control Panel**).

Note There are limitations as to the behavior of certain macro triggers when working with documents on QlikView Server. Please study the special section "Considerations when using macros in QV documents on the QlikView Server" in the QlikView Server Reference Manual before designing server documents with macro triggers.

Font

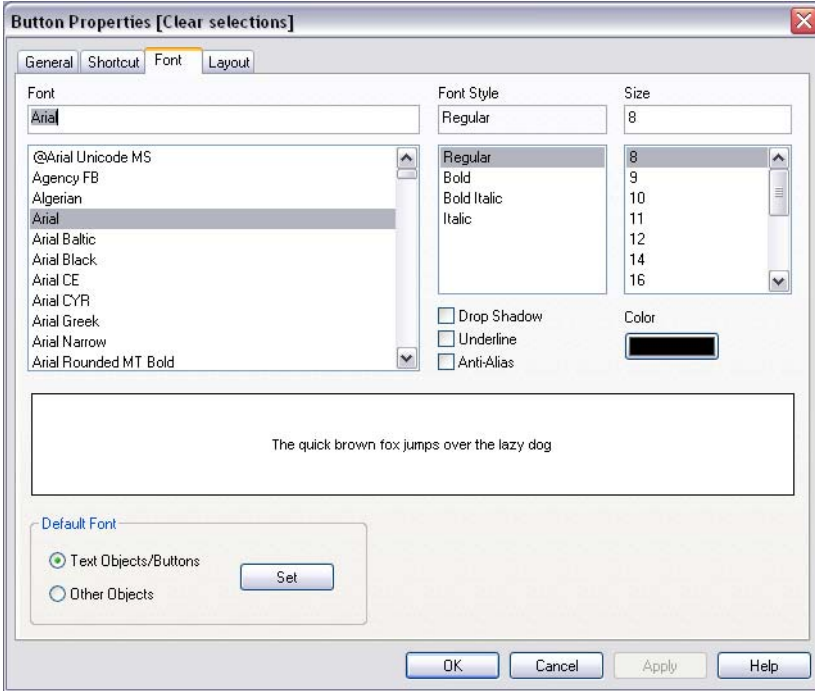


Figure 63. The Font page in the Button Properties dialog

In the **Font** page, you can specify the font, style and size of the button text. In addition, for buttons it is also possible to specify a color for the button text by using the **Color** settings.

Color

This button opens the **Color Area** dialog (see page 23) where the font color can be set.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the button should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are set-

tings for multi-line text and text alignment in both vertical and horizontal direction.



12 THE TEXT OBJECT

Text objects can be used for displaying text information or an image in the layout.

12.1 Creation

Text objects are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu or by clicking the **Create Text** button in the toolbar.



12.2 Object Menu of the Text Object

The text object Object menu is found as the Object menu when a text object is active. It can also be opened as a context menu by clicking with the right mouse button on a text object. The menu contains the following commands:

Properties...

Opens the **Text Object Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Print...

Opens the standard **Print** dialog box (see page 263), allowing you to print the text object. Use CTRL+P to open this dialog with the keyboard shortcut.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send to Excel

Exports the text to Microsoft Excel, which is automatically launched if not already running. The text will appear in a single cell in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer.

Copy to Clipboard

This menu contains the various copy options for the sheet object.

Text

Copies the text displayed in the text object to the clipboard.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the text object from the sheet.

12.3 Text Object Properties

The **Text Object Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

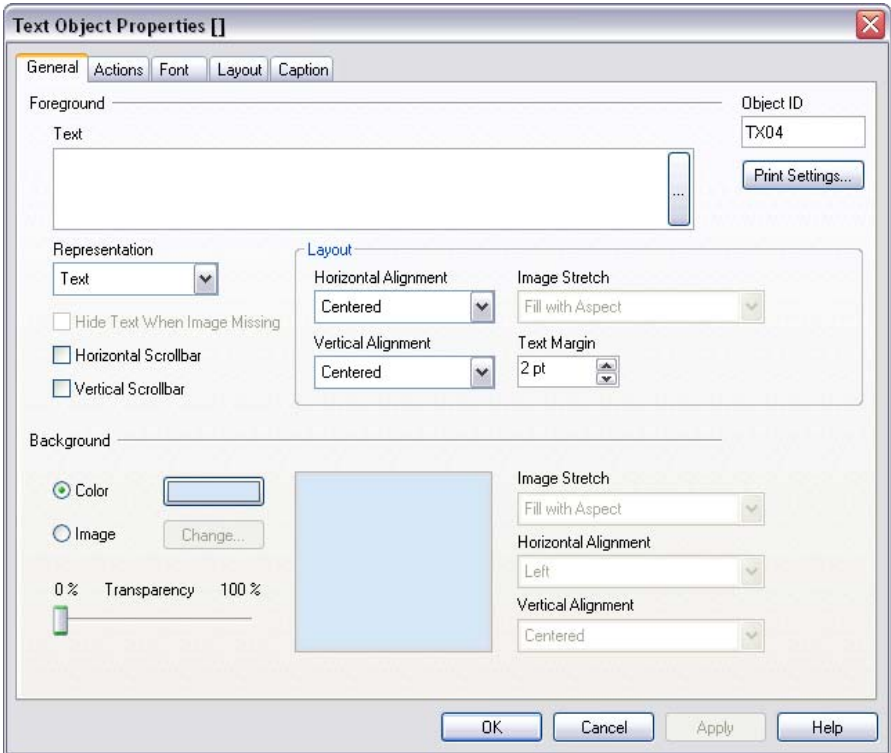


Figure 64. The General page in the Text Object Properties dialog

On the **General** page settings for the text, its background, alignment etc. can be made.

Foreground

In the **Foreground** group you can specify the foreground of the text object. The foreground usually consists of the text itself, but it may also be an image.

Text

This is where you enter the text to be displayed. The text may be defined as a calculated formula (see page 467) for dynamic update.

Representation

The text in the Text object may be interpreted as a reference to an image in memory or on disk. When selecting **Text** the

contents of the Text object will always be interpreted and displayed as text. When selecting **Image** QlikView will try to interpret the text contents as a reference to an image. The reference may be a path to an image file on disk (e.g. `c:\MyPic.jpg`) or inside the .qvw document (e.g. `qmem://<Name>/<Peter>`). It can also be an info function linked to a field containing image info (e.g. `=info(MyField)`). If QlikView cannot interpret the text contents as a valid reference to an image, the text itself will be displayed.

Hide Text When Image Missing

If this option is selected, QlikView will not display text in the text object if interpretation as an image reference fails for some reason. The text object will then be left blank.

Horizontal Scroll Bar

If this check box is marked a horizontal scroll bar will be added to the text object when its text content is too wide to be displayed within the given area.

Vertical Scroll Bar

If this check box is marked a vertical scroll bar will be added to the text object when its text content is too long to be displayed within the given area.

Layout

In this group you define how QlikView should display the text or foreground image within the text object area.

Horizontal Alignment

The text or image can be horizontally left-aligned, centered or right-aligned within the text object.

Vertical Alignment

The text or image can be vertically top-aligned, centered or bottom-aligned within the text object.

Image Stretch

This setting describes how QlikView formats the foreground image to fit in the text object area. There are four alternatives.

No Stretch

If this option is selected, the foreground image will be shown as is without any stretching. This may cause parts of the picture to be invisible or only part of the text object to be filled.

Fill

If this option is selected, the foreground image will be stretched to fit the text object area without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the foreground image will be stretched as far as possible to fill the text object area while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the foreground image will be stretched to fill the text object area in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Text Margin

With this setting you may create a margin between the outer borders of the text object and the text itself. The width of the margin can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Background

In the **Background** group you can specify the background of the text object. The current settings are reflected in the preview pane to the right.

Color

Select this radio button if you want the text to be displayed on a colored background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Image

Select this radio button if you want to display an image as background. To change the current image, click the **Change** button. If no text is entered in the **Text** field, the text object can be used for displaying a fixed image as a sheet object in the layout. Image types supported include jpg, png, bmp, gif and animated gif.

Transparency

Sets the degree of transparency of the text object background. At 100% the background will be completely trans-

parent. The transparency will apply regardless if a color or an image is used for background.

Image Stretch

This setting describes how QlikView formats the background image to fit in the text object area. There are four alternatives.

No Stretch

If this option is selected, the background image will be shown as is without any stretching. This may cause parts of the picture to be invisible or only part of the text object to be filled.

Fill

If this option is selected, the background image will be stretched to fit the text object area without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the background image will be stretched as far as possible to fill the text object area while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the background image will be stretched to fill the text object area in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Horizontal Alignment

The text can be horizontally left-aligned, centered or right-aligned within the text object.

Vertical Alignment

The text can be vertically top-aligned, centered or bottom-aligned within the text object.

Object ID

The unique ID of the current text object. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first text object of a document will be assigned the ID TX01. An object ID may be changed to any other

string that is currently not used for any other sheet object, sheet or bookmark in the document.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Actions

In the **Actions** page you can specify what actions should be performed when you click on the object. The page is identical to the **Actions** page for the **Button Object**, see page 200 in Book II.

Font

In the **Font** page (Figure 33 on page 113), you can specify the font, style and size of the text in the text object.

It is also possible to specify a color for the text by using the settings in the **Color** group.

Color

Select this radio button if you want the object to be displayed with a colored background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the text object should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.

13 THE LINE/ARROW OBJECT

The Line/Arrow Object is a sheet object that draws a line or an arrow in the layout. A line can, for example, make the layout clearer by separating the documents into sections.

13.1 Creation

Line/Arrow Objects are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu or by pressing the **Create Line/Arrow Object** button in the toolbar.



13.2 Object Menu of the Line/Arrow Object

The Line/Arrow object **Object** menu is found as the **Object** menu when a Line/Arrow Object is active. It can also be opened as a context menu by clicking with the right mouse button on a Line/Arrow Object.

Properties...

Opens the **Line/Arrow Object Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Copy to Clipboard

This menu contains the various copy options for the sheet object.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the current Line/Arrow Object from the sheet.

13.3 Line/Arrow Object Properties

The **Line/Arrow Object Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

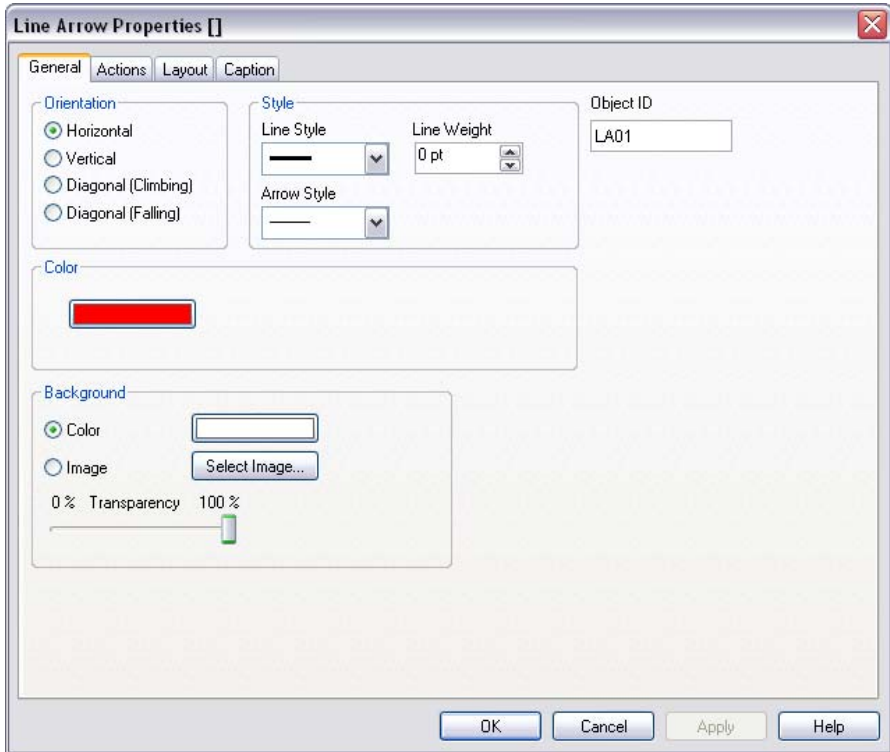


Figure 65. The General page in the Line/Arrow Object Properties dialog

On the **General** page of the **Line/Arrow** Object general settings (e.g. attributes and background) for line/arrows can be set.

Orientation

Sets the direction of the line/arrow in horizontal, vertical or one of two diagonal modes.

Style

In this group the formatting of the line/arrow can be defined.

Line Style

This drop-down box is used to set the line to solid, dashed or dotted mode.

Arrow Style

This drop-down box is used to decide whether arrowheads are to be drawn at one or both ends of the line and whether arrowheads should be filled or not.

Line Weight

Sets the weight of the line. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Color

Click the button to specify a color for the line/arrow.

Background

In the **Background** group you can specify the background of the line/arrow object.

Color

Select this radio button if you want the object to be displayed with a colored background. The color can be defined as a solid color or a gradient via the Color Area dialog that opens when clicking the button.

Image

Select this radio button if you want to display an image as background. To change the current image, click the **Select Image** button.

Transparency

Sets the degree of transparency of the line/arrow object background. At 100% the background will be completely transparent. The transparency will apply regardless if a color or an image is used for background.

Object ID

The unique ID of the current line/arrow object. Upon creation, every layout entity is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of entity, as well as of a number. The first line/arrow object of a document will be assigned the ID LA01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet, alert, bookmark or report in the document.

Actions

In the **Actions** page you can specify what actions should be performed when you click on the object. The page is identical to the **Actions** page for the **Button Object**, see page 200 in Book II.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the Line/Arrow Object should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.



14 THE SLIDER/CALENDAR OBJECT

The Slider/calendar object provides alternative means to select values in QlikView fields or enter values in QlikView variables. The slider/calendar object has, as the name suggests, two completely different modes. As different as they may look, they still work very much the same way behind the visual user interface.

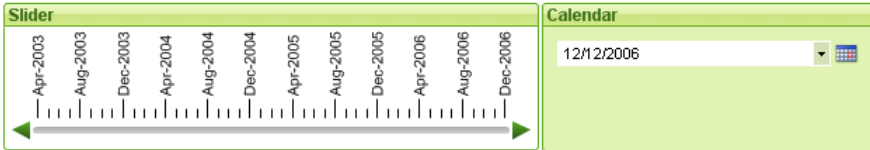


Figure 66. The two main incarnations of a slider/calendar object side by side, a slider to the left and a calendar input control to the right.

14.1 Using a Slider/Calendar Object

Slider Mode

A single value, two values or a range of values (depending on the properties of the object) is selected in a field or inserted into one or two variables by means of dragging a thumb tack along the slider background. The thumb tack can also be moved by means of scroll arrows. Some sliders may incorporate a scale and tick marks for navigation. See Figure 70 below for a graphical explanation of the various parts of a slider.

Calendar Mode

A slider/calendar object in calendar mode appears in the layout as a drop-down box with a calendar icon to the right. When clicking on the calendar icon it will expand to a calendar control (see below). It is possible to navigate between months and years by means of the arrow buttons or the month and year drop-down controls. After selecting a date or a range of dates (depending on the properties of the object) in the calendar control the selection will be applied to the underlying field or value.

If the object is connected to a field, selections in the field will be reflected in the expanded calendar control, using the same selection color codes as found in list boxes (e.g. green for selected values). Provided that the object is configured for multiple values, it will be possible to paint over multiple values just like in a list box. By pressing the CTRL key it is even possible to select multiple ranges of values, even if they appear in different months or years.

Once the calendar control is closed after the selection, the drop-down box will function essentially as a field drop-down in a multi box.

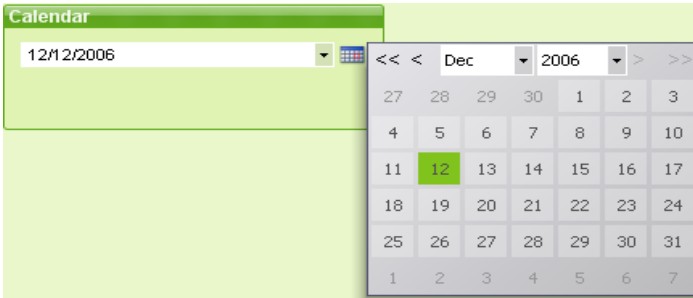


Figure 67. A calendar input control in an open state

14.2 Creation

Slider/calendar objects are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu, or by pressing the **Create Slider/Calendar Object** button in the toolbar.

14.3 Object Menu of the Slider/Calendar Object

The Slider/Calendar object object menu is found as the Object menu when a Slider/Calendar object is active. It can also be opened as a context menu by clicking with the right mouse button on a Slider/Calendar Object. The menu contains the following commands:

Properties...

Opens the **Slider/Calendar Object Properties** dialog. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Copy to Clipboard

This menu contains the various copy options for the sheet object.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.



Minimize

Iconizes the slider/calendar object. Clicking  in the object caption (if shown) gives the same result. To restore the object to its previous size and location, double-click the minimized object. This command is available only if you have selected the option **Allow Minimize** on the **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized slider/calendar object to its previous size and location. It is also possible to restore a minimized object by double-clicking the caption or clicking , provided that the caption is shown. This command is available only if you have selected at least one of the options **Allow Minimize** or **Allow Maximize** on the **Caption** page (see Figure 36 on page 120) and if the object is in the minimized or maximized state.

Maximize

Enlarges the slider/calendar object to fill the sheet. Clicking  in the object caption (if shown) gives the same result. To restore the slider/calendar object to its previous size and location, choose **Restore** or click . This command is available only if you have

selected the option **Allow Maximize** on the **Caption** page (see Figure 36 on page 120).

Help

Opens context-specific help.

Remove

Removes the active Slider/Calendar object from the sheet.

14.4 Slider/Calendar Object Properties

The **Slider/Calendar Object Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

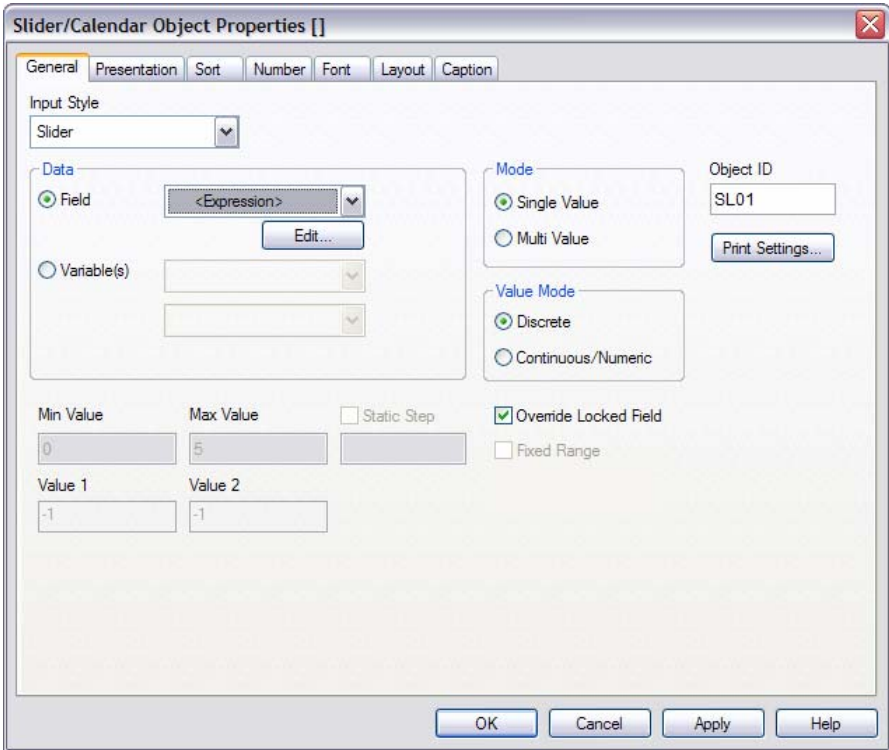


Figure 68. The General page of the Slider/Calendar Object Properties dialog

In the **General** page settings for the data to be manipulated by the Slider/Calendar Object can be made.

Input style

In this drop-down you select the basic mode of the slider/calendar object.

Slider

Select this option if you want the object to be represented by a slider control.

Calendar

Select this option if you want the object to be represented by at calendar input control.

Data

In this group you define which data object the slider/calendar object should be connected to. A slider/calendar object can be connected either to a field or to one or two variables.

Field

Select this radio button if you want to connect the slider/calendar object to a field or an expression. Select the field in the drop-down box. If **Expression** is chosen in the drop-down the **Edit Expression** dialog will open.

Edit...

Opens the **Edit Expression** dialog for the expression chosen in the drop-down.

Variable(s)

Select this radio button if you want to connect the slider/calendar object to one or two variables. Select the variable(s) in the drop-down box(es). The second variable is only available if you select **Multi Value** under **Mode** below.

Mode

In this group you specify if the slider/calendar object should be used to select a single value or a range of values.

Single Value

Select this radio button for the slider/calendar object to select a single value in a field or set a value to a single variable.

Multi Value

Select this radio button for the slider/calendar object to select a range of values in a field or set a value to each of two separate variables.

Value Mode

In this group you specify if the slider/calendar object should select discrete values or define a continuous range of values. This group is not available in calendar mode.

Discrete

Select this radio button for the slider/calendar object to map to existing field values or predefined variable values. This option is not available for **Multi Value** mode with **Variable(s)** data. This group is always assumed for calendar mode.

Continuous/Numeric

Select this radio button for the slider/calendar object to map to all field values within a continuous numeric range or pre-defined variable values. This option is not available for **Single Value** mode with **Field** data.

Min Value

Sets the minimum value for the slider/calendar object in **Continuous/Numeric** value mode.

Max Value

Sets the maximum value for the slider/calendar object in **Continuous/Numeric** value mode.

Static Step

Mark the check box and enter a value in the edit box to specify a static step for slider/calendar object values in the **Continuous/Numeric** value mode. This group is not available in calendar mode.

Value 1

Shows current value for the slider/calendar object in **Continuous/Numeric** value mode if slider/calendar object is **Continuous/Numeric**. In **Continuous/Numeric** mode the low value is shown.

Value 2

Shows current high value for the slider/calendar object in **Continuous/Numeric** value mode. Only relevant in **Multi Value** mode.

Override Locked Field

If this check box is marked, selections can be made via the slider/calendar object in a field even if the field is locked. the field will still be locked for logical changes stemming from selections in other fields. This option is selected by default for slider/calendar objects.

Fixed Range

If this check box is marked, the width of the numeric range defined for a slider/calendar object in **Continuous/Numeric** and **Multi Value** mode will be locked. The user will then not be able to widen or narrow the range by dragging the edges of the slider thumbtack.

Print Settings...

By pressing this button you will gain access to the **Print Settings** dialog where margins and header/footer format can be defined. The **Print Settings** dialog holds two pages, **Layout** and **Header/Footer**, which are identical to the last two pages of the **Print** dialog (see page 263).

Object ID

The unique ID of the current slider/calendar object. Upon creation, every layout entity is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of entity, as well as of a number. The first slider/calendar object of a document will be assigned the ID SL01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet, alert, bookmark or report in the document.

Presentation (Slider Mode)

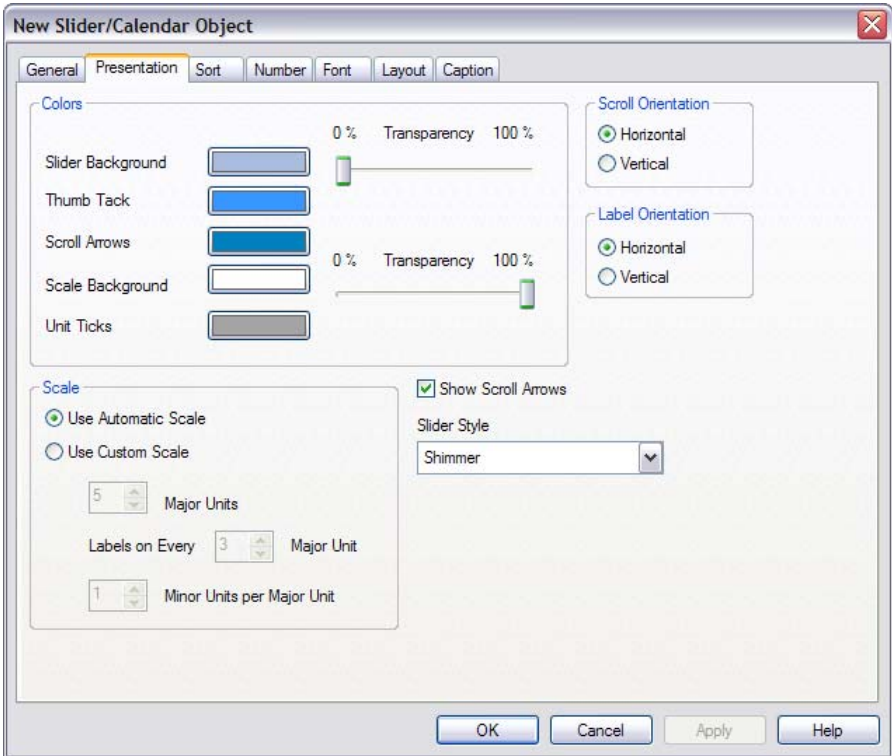


Figure 69. The Presentation page in the Slider Object Properties page.

In the **Presentation** page for slider mode the visual properties of the slider control can be configured.

Colors

In this group you can define colors for the different parts of the slider (see Figure 70 on page 233).

Slider Background

Defines the color of the slider background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button. The degree of transparency of the slider background can be set using the **Transparency** control to the right of the button. At 0% the background will be completely opaque. At 100% the background will be completely transparent.

Thumbtack

Defines the color of the slider thumbtack. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Scroll Arrows

Defines the color of the slider scroll arrows, if shown. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Scale Background

Defines the color of the slider scale background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button. The degree of transparency of the scale background can be set using the **Transparency** control to the right of the button. At 0% the background will be completely opaque. At 100% the background will be completely transparent.

Ticks

Defines the color of the slider scale ticks and text.

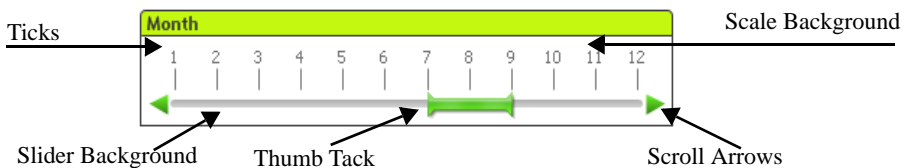


Figure 70. The different parts of a slider/calendar object.

Scroll Orientation

The slider can be oriented in **Horizontal** or **Vertical** direction.

Label Orientation

Sets whether scale text should be oriented in **Horizontal** or **Vertical** direction.

Scale

In this group you can define if the slider should have a scale and how it is to be formatted.

Use Automatic Scale

If this option is selected scale settings will be automatically based on slider size and underlying values.

Use Custom Scale

If this option is selected scale settings can be set manually using the three settings described below.

Major Units

Defines the number of major units on the slider scale.

Label On Every n Major Unit

Defines the density of scale label text.

Minor Units Per Major Unit

Defines the number of minor units between each major unit on the slider scale.

Show Scroll Arrows

Mark this check box if you want scroll arrows to be shown in the slider arrows.

Slider Style

Here you may select different looks for the slider.

Presentation (Calendar Mode)

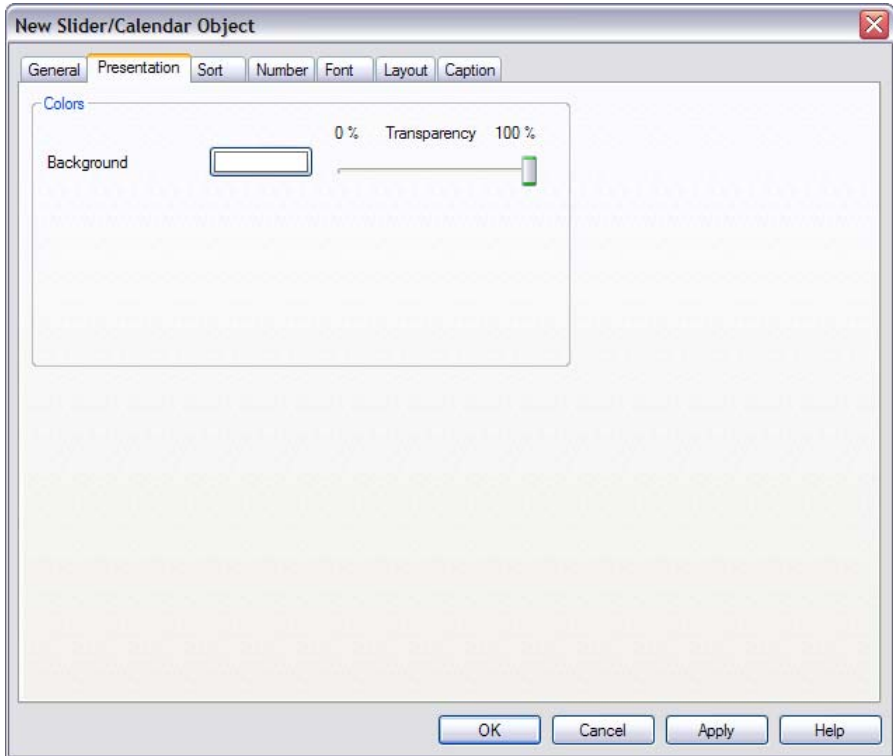


Figure 71. The Presentation page for the calendar mode

In the **Presentation** page for calendar mode the visual properties of the calendar control can be configured.

Colors

In this group you can define colors for the calendar control.

Background

Defines the color of the calendar control background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button. The degree of transparency of the slider background can be set using the **Transparency** control to the right of the button. At 0% the background will be completely opaque. At 100% the background will be completely transparent.

Sort

On the **Sort** page the sort order can be set for the slider data. This data is very similar to its counterpart under **Document Properties** (see page 54).

Number

On the **Number** page the number format can be set for the slider scale. This dialog is very similar to its counterpart under **Document Properties** (see page 58).

Font

On the **Font** page (page 113), you can change the font of the text in the Slider/Calendar Control Object.

You can specify the font, style and size of the text. The font chosen applies to all text contained in the Slider/Calendar Control object.

Layout

On the **Layout** page (page 114), you can specify how the Slider/Calendar Control Object should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.

15 THE BOOKMARK OBJECT

The bookmark object is a sheet object that is used to display and select selection bookmarks. Depending on its configuration it can also be used for adding new bookmarks or deleting old ones. The bookmark object basically offers the same options as the **Bookmark menu** (page 69 in Book I) but can be positioned on the sheet as a sheet object.

15.1 Creation

Bookmark objects are created choosing **New Sheet Object** from the **Layout** menu or from the sheet's **Object** menu, or by pressing the **Create Bookmark Object** button in the toolbar.

15.2 Object Menu of the Bookmark Object

The bookmark object menu is found as the Object menu when a bookmark object is active. It can also be opened as a context menu by clicking with the right mouse button on a bookmark object. The menu contains the following commands:

Properties...

Opens the **Properties** dialog of the Bookmark object. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Add Bookmark

Saves the current set of selections as a bookmark. The default name can be changed in the **Create Bookmark dialog** that opens automatically. This command can also be invoked via the keyboard shortcut CTRL+B.

Remove Bookmark

Opens a menu with the ten most recently used bookmarks defined in the document. By choosing one of them, that bookmark will be removed from the document.

Replace Bookmark

Opens a menu with the ten most recently used bookmarks currently defined in the document. By choosing one of these, the contents of that bookmark will be replaced with the current state of selections and variable values.

Import Bookmarks...

Opens the **Import Bookmark(s)** dialog for importing bookmarks from a QlikView bookmark file.

Export Bookmarks...

Opens the **Export Bookmark(s)** dialog for exporting bookmarks to a QlikView bookmark file.

Copy to Clipboard

This menu contains the various copy options for the sheet object.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the active bookmark object from the sheet.

15.3 Bookmark Object Properties

The **Bookmark Object Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed you probably don't have the privileges needed to perform property changes (see page 85). When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

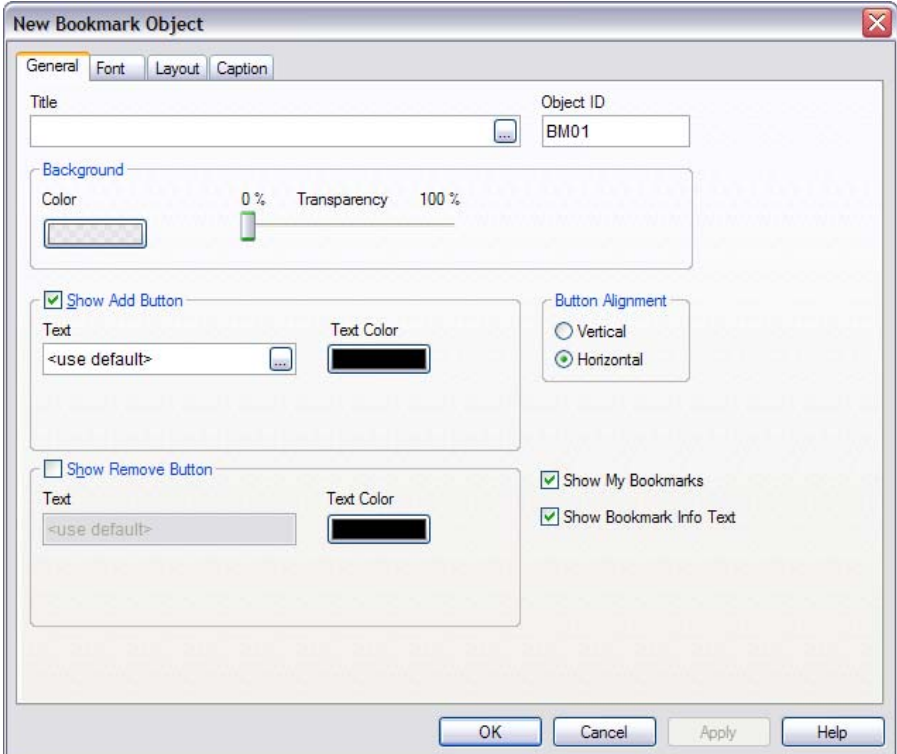


Figure 72. The General page in the Bookmark object Properties dialog.

In the **General** page settings for title and configuration of the Bookmark object can be made.

Title

Sets the title for the bookmark object. This name will be shown in the object caption. The title may be defined as a calculated formula (see page 467 in Book III) for dynamic update.

Object ID

The unique ID of the current bookmark object. Upon creation, every layout entity is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of entity, as well as of a number. The first bookmark object of a document will be assigned the ID BM01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet, alert, bookmark or report in the document.

Background

In this group you specify the appearance of the bookmark object background.

Color

Sets the background color of the bookmark object. The color can be defined as a solid color or a gradient via the **Color Area dialog** that opens when clicking the **Color** button.

Transparency

Sets the degree of transparency of the bookmark object background. At 0% the background will be completely opaque with the color defined under **Background Color** above.

Show Add Button

In this group you set the options for an **Add bookmark** button in the bookmark object.

Text

Here you may type a text to be shown on the **Add Bookmark** button. If no text is typed the edit box will show <use default> and the text “Add Bookmark” will be shown on the button.

Text Color

Sets the text color of the **Add Bookmark** button.

Show Remove Button

In this group you set the options for a **Remove Bookmark** button in the bookmark object.

Text

Here you may type a text to be shown on the **Remove Bookmark** button. If no text is typed the edit box will show <use default> and the text “Remove Bookmark” will be shown on the button.

Text Color

Sets the text color of the **Remove Bookmark** button.

Button Alignment

Here you may choose how the **Add** and **Remove** buttons are to be positioned in relation to each other, when both are shown.

Vertical

The buttons are positioned on top of each other.

Horizontal

The buttons are positioned side by side.

Show My Bookmarks

Mark this check box in order to display personal bookmarks in the bookmark object list of bookmarks. Personal bookmarks will be listed last with a divider separating the document bookmarks from the personal bookmarks.

Show Bookmark Info Text

Mark this check box to display the text entered for the bookmark under **Edit Info** in the **Bookmarks** dialog, **Document Bookmarks** page.

Font

On the **Font** page (Figure 33 on page 113), you can change the font of the text in the **Bookmark** object.

You can specify the font, style and size of the text. The font chosen applies to all text contained in the bookmark object.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the bookmark object should appear in the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction.

16 THE SEARCH OBJECT

Search objects can be used for searching for information anywhere in the document.

16.1 Creation

Search objects are created by choosing **New Sheet Object** from the **Layout** menu or from the sheet **Object** menu.



16.2 Object Menu of the Search Object

The Search Object Object menu is found as the Object menu when a search object is active. It can also be opened as a context menu by clicking with the right mouse button on a search object. The menu contains the following commands:

Properties...

Opens the **Search Object Properties** dialog box. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Copy to Clipboard

This menu contains the various copy options for the sheet object.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the sheet object from the sheet.

16.3 Search Object Properties

The **Search Object Properties** dialog is opened by choosing **Properties** from the **Object** menu. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

General

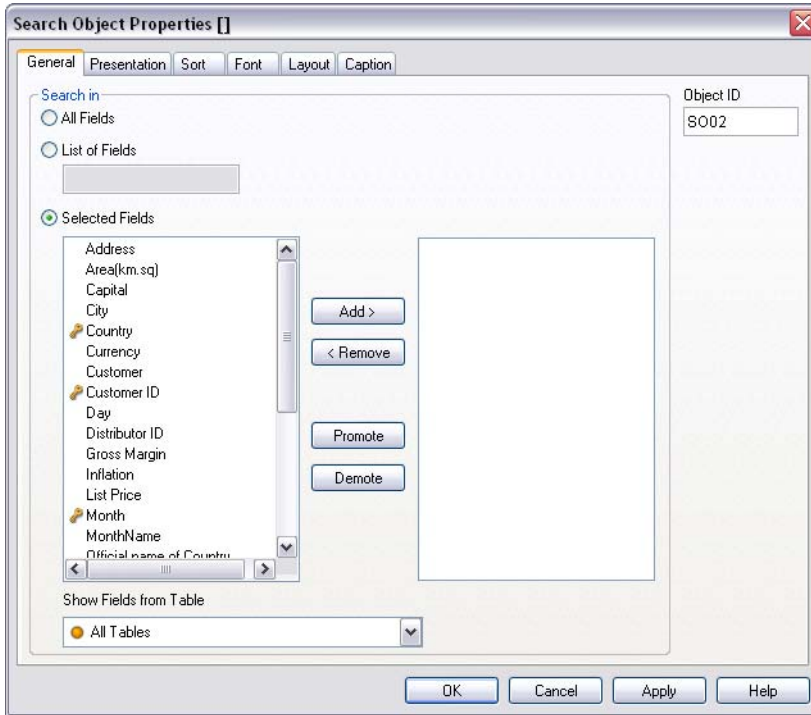


Figure 73. The General page in the Search Object Properties dialog

On the **General** page settings you can configure the search options.

Search in

All Fields

Mark this radio button to allow the search object to search all fields in the document.

List of Fields

Mark this radio button and list the fields in the field below to set which fields should be searchable. Use semicolon (;) as separator between fields. The wildcard characters * and ? are allowed in field names. The list may be defined as a calculated formula (see page 467) for dynamic update.

Selected Fields

Mark this radio button and choose what fields should be searchable.

Show Fields from Table

This drop-down menu shows the value **-All Tables-** and the **Available Fields** list contains all fields of the document. By selecting a specific table name in the drop-down menu you may limit the **Available Fields** list to only fields from that input table.

Finally, you may select the **-All Tables (Qualified)-** which will show all fields in the document qualified with their table name. A field will appear one time for each table it appears in.

Object ID

The unique ID of the current search object. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first search object of a document will be assigned the ID SO01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet or bookmark in the document.

Presentation

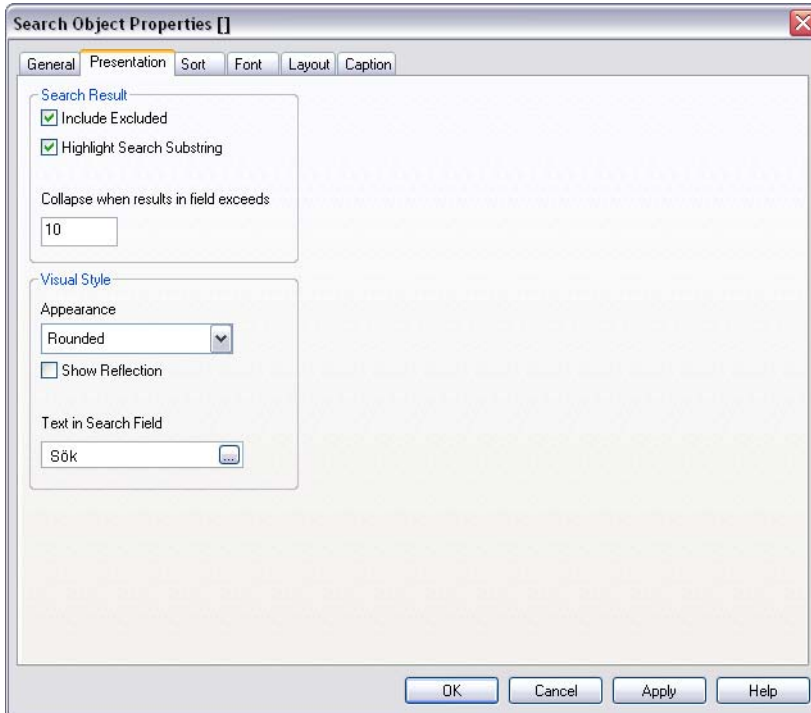


Figure 74. The Presentation page in the Search Object Properties dialog

Search Result

Include Excluded

Mark this check box to include those values in the search that have been excluded by selections.

Highlight Search Substring

The search string so far will be highlighted in the matches.

Collapse When Results in Field Exceeds

Limit the number of displayed matches in each field.

Visual Style

Appearance

Choose between a **Rounded** and **Squared** look.

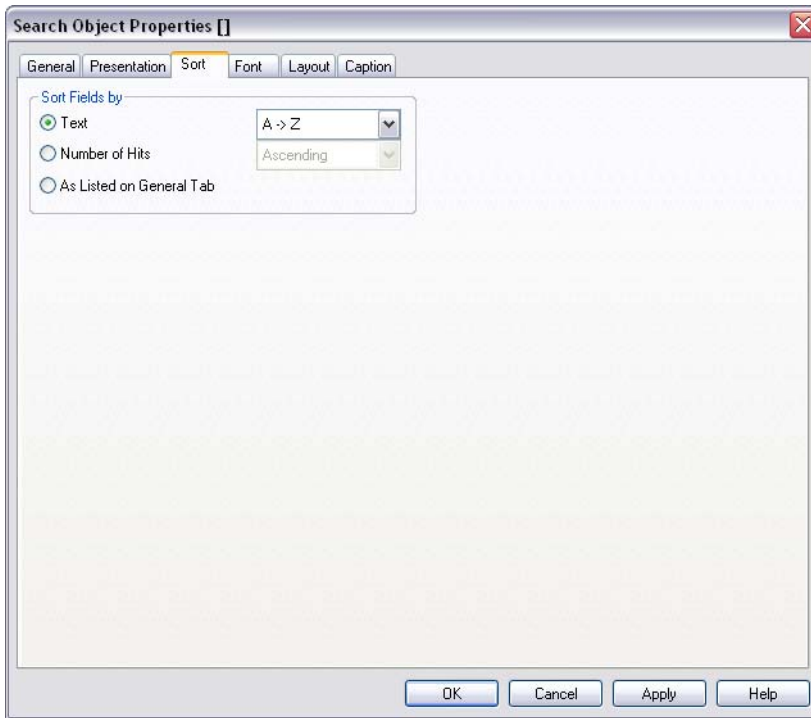
Show Reflection

A reflection of the search object will be shown in the layout.

Text in Search Field

Enter the text that should be visible in the search object before you enter a search string. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Sort



The Sort page in the Search Object Properties dialog

In this group, the sort order of the fields in the search object can be set.

Text

Sorts the fields containing search hits in alphabetical order.

Number of Hits

Sorts the fields containing search hits according to the number of hits in each field.

As Listed on General Tab

Sorts the fields containing search hits according to the order in which the fields are listed on the **General** tab.

Font

In the **Font** page (Figure 33 on page 113), you can specify the font, style and size of the text in the sheet object.

It is also possible to specify a color for the text by using the settings in the **Color** group.

Color

Select this radio button if you want the object to be displayed with a colored background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Layout

On the **Layout** page (Figure 34 on page 114), you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.



17 CUSTOM OBJECT

The **Custom Object** is a sheet object that is used to carry custom defined OCX replacement controls. Replacement controls are windowless OCX controls programmed by QlikTech, yourself or third party vendors and which comply with the specifications for QlikView replacement OCX controls. The OCX control will be drawn in the rectangle on the sheet outlined by the underlying custom object. Communication between the OCX control and the QlikView document is maintained via the QlikView Automation interface.

The custom object adds the possibility to extend QlikView layouts with functionality not found in the product itself. When designing QlikView a strong focus has always been given on only adding generally applicable functionality. The OCX replacement control is a way to allow more specific sheet objects suited for special underlying data sets or display needs.

The QlikView installation DVD comes with a few sample OCX replacement controls. These are provided as is and with full VB source code. You may use them as they are or modify the source code to better suit your needs. QlikTech does not offer support on these sample controls.

17.1 Creation

Custom Objects are created by choosing **New Sheet Object** from the **Layout** menu or from the **Sheet object** menu, or by pressing the **Create Custom Object** button in the toolbar.

17.2 Object Menu of the Custom Object

The **Object** menu of the **Custom Object** is found as the **Object** menu when a **Custom Object** is active. It can also be opened as a context menu when right-clicking a **Custom Object**. The menu contains the following commands:

Properties for Replacement Control...

Opens the **Properties** dialog for the OCX replacement control used. This dialog is programmed by the provider of the OCX replacement control and may vary in looks and functionality. It may also be unavailable for certain replacement controls.

Properties...

Opens the **Custom Object Properties** dialog. Use ALT+RETURN to open this dialog with the keyboard shortcut.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Copy to Clipboard

This menu contains the various copy options for the sheet object.


Image

Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the sheet entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.

Minimize


Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Maximize

Enlarges the object to fill the sheet. Clicking  in the object caption (if shown) produces the same result. This command is available

only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 36 on page 120).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the active custom object from the sheet.

17.3 Custom Object Properties

The Custom Object Properties dialog is opened by choosing Properties from the Object menu. If the Properties command is dimmed you probably don't have the privileges needed to perform property changes (see page 85). When the desired prop-

erties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

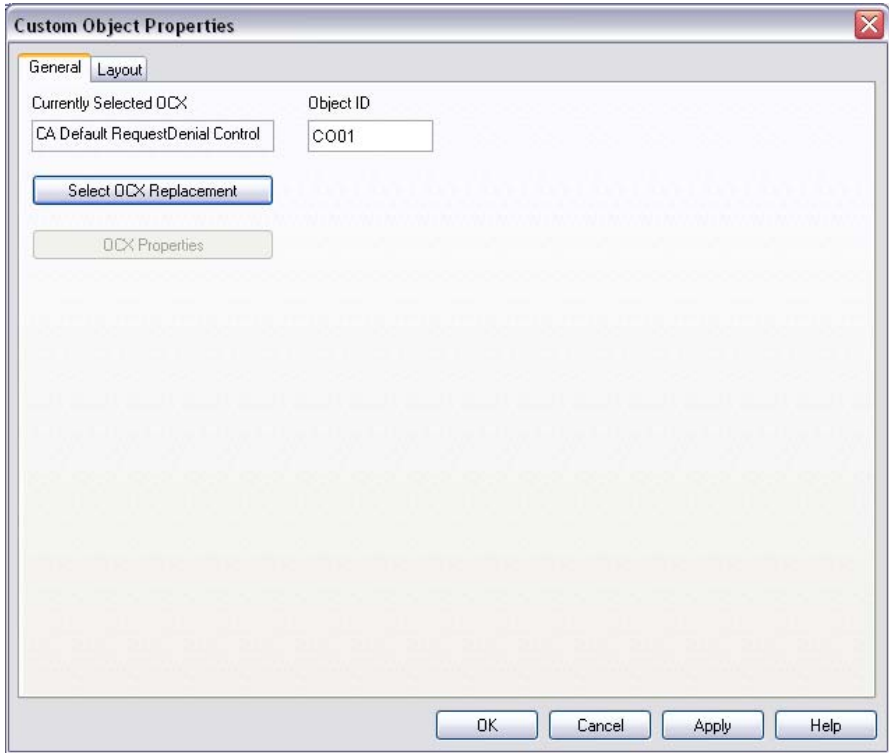


Figure 75. The General page in the Custom Object Properties dialog

In the **General** page settings for title and configuration of the Custom Object can be made.

Currently Selected OCX

Shows the name of the currently selected OCX replacement.

Object ID

The unique ID of the current custom object. Upon creation, every layout entity is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of entity, as well as of a number. The first custom object of a document will be assigned the ID CO01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet, alert, bookmark or report in the document.

Select OCX Replacement

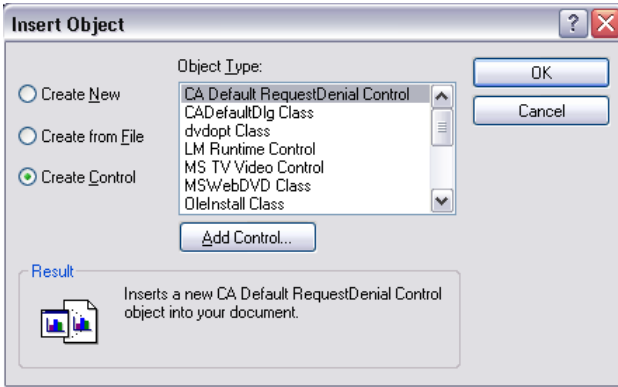


Figure 76. The Insert Object dialog

Press this button to open the Insert Object dialog where you may select or change an OCX replacement control. The Object Type list contains all insertable OCX controls registered on your system. Select one OCX in the list and click **OK**.

OCX Properties

Opens the **Properties** dialog for the OCX replacement control used. This dialog is programmed by the provider of the OCX replacement control and may vary in looks and functionality. It may also be unavailable for certain replacement controls.

Layout

On the **Layout** page (Figure 34 on page 114) you can specify how the custom object should appear on the layout. This includes settings for shape, border and the layer on which it should reside.

Caption

On the **Caption** page (Figure 36 on page 120), advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page.



18 SERVER COLLABORATION

18.1 The Collaboration Pane

The **Collaboration Pane** offers an easy way for end-users to manage their own server sheet objects and to access other users' shared objects.

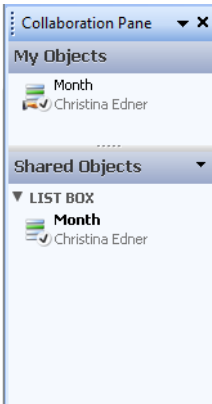


Figure 77. The Collaboration Pane

Configuring the pane

The **Collaboration Pane** can be toggled on and off via the **Collaboration Pane** command in the **View** menu, or by pressing F2. The pane can be moved and docked to any side of the QlikView application window. It can also be left free-floating next to the QlikView window.

The **Collaboration Pane** can be left on when working with local documents (non server-based) but then it has no function.

The **Collaboration Pane** contains two areas. The **My Objects** area lists your own personal server objects for the active server document. The **Shared Objects** area lists all shared sheet objects (including your own shared objects) linked to the current server document.

To hide an object in the QlikView application window, right-click the object and choose **Hide**. In order to show the object in the layout again you must drag the object from the **Collaboration Pane**.

When hovering an object, a pop-up is shown containing information about the name of the object, the object type, owner and latest modification date.

Shared objects

Here the shared objects are shown. They can be grouped differently according to **Type**, **Owner** and **Date**. Change the grouping by clicking on the arrowhead next to shared objects. The list can be expanded to show more information about each object or compressed to save space. Click on the arrowhead next to the object to expand, click again to compress.

The objects that are shown in the layout are marked by a check in the **My Objects** and **Shared Objects areas**.

To use someone else's shared object you must drag it to the application window.

Adding and sharing objects

When you create a new server sheet object it will automatically be added to the **My Objects** list.

To share one of your own server objects to other users, right-click on the object in the **My Objects** area and choose **Share with Everyone** or **Share Permissions...** Choosing the latter will open the dialog below.

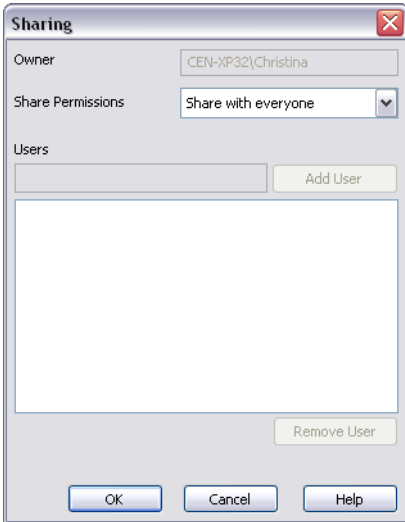


Figure 78. The Sharing dialog of the Collaboration Pane

In this dialog you can configure how the object should be shared. To share an object choose one of the options in the **Share Permissions** drop-down menu.

Do Not Share

The object is not shared with any users.

Share with Everyone

The object is shared with all users.

Share by Username

The object is shared with the users that are listed below.

The object will appear in the **Shared Objects** area but also remain in the **My Objects** area, now marked with a small hand to indicate its shared status.

To unshare an object which you have previously shared, right-click on the object in the **My Objects** area, choose **Unshare** or **Share Permissions** again to open the **Sharing** dialog and then choose **Do Not Share** in the drop down **Share Permissions**. The object will disappear from the **Shared Objects** list and no longer be available for other users. Note that it will however not disappear from other users' active sessions).

18.2 The Server Objects Dialog

The **Server Objects** dialog offers another more list-oriented way to manage your personal server objects and to select among other users' shared server objects. The dialog is opened from the Layout menu and contains two pages:

My Server Objects

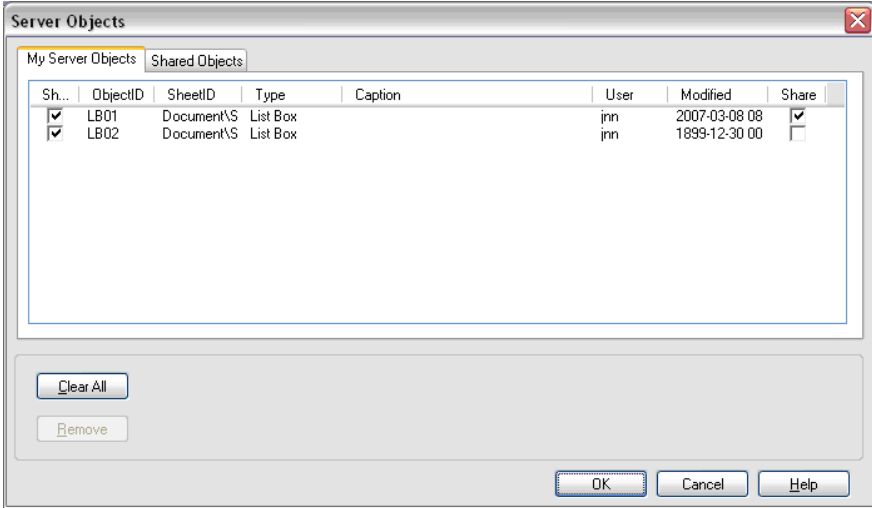


Figure 79. The My Server Objects page of the Server Objects dialog

In the **My Server Objects** page you can manage your own personal server objects and share them with other users.

At the top of the dialog you will find a list of all personal server objects currently defined for the QlikView document. The objects are described by, and may be sorted by several columns:

Show

Mark this check box if you want the object to appear in the layout of the document. If the check box is left unchecked, the object will not appear, but will remain accessible via this dialog.

Object ID

The unique ID of the object.

Sheet ID

The ID of the sheet where the object will reside when shown.

Type

The type of the object (e.g. list box).

Caption

The caption of the object (if defined).

User

The Windows identity of the user who created the object (yourself).

Modified

A timestamp indicating when the object was last modified.

Share

This check box is only available in the **My Server Objects** page. By marking it for one of your personal server objects, you will make that object available to other authenticated users of the same server document in their **Shared Objects** page. The object will stay in your **My Server Objects** page and will not appear in your **Shared Objects** page. You may at any time revoke the sharing by deselecting the check box.

At the bottom of the dialog you will find buttons which can be used to perform actions with the objects in the list above.

Clear All

Removes all objects in the list.

Remove

Removes the selected object from the list.

Shared Objects

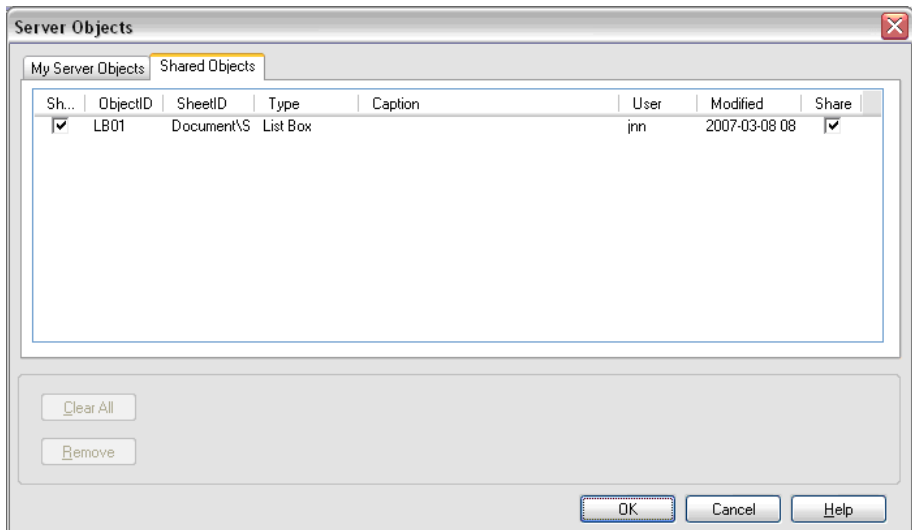


Figure 80. The Shared Objects page of the Server Objects dialog

In the **Shared Objects** page you can manage the display of other users' shared server objects.

At the top of the dialog you will find a list of all shared server objects currently defined for the QlikView document. The objects are described by, and may be sorted by several columns:

Show

Mark this check box if you want the object to appear in the layout of the document. If the check box is left unchecked, the object will not appear, but will remain accessible via this dialog.

Object ID

The unique ID of the object.

Sheet ID

The ID of the sheet where the object will reside when shown.

Type

The type of the object (e.g. list box).

Caption

The caption of the object (if defined).

User

The Windows identity of the user who created the object.

Modified

A timestamp indicating when the object was last modified.

Share

This check box is only available in the **My Server Objects** page.

At the bottom of the dialog you will find buttons which can be used to perform actions with the objects in the list above.

Clear All

Removes all objects in the list.

Remove

Removes the selected object from the list.

19 EXPORTING AND PRINTING

19.1 Print

To open this dialog box, choose **Print...** or **Print Possible...** from the **File** menu, from the toolbar or from the **Object** menu of a printable object. The **Print** dialog box contains three pages: **General**, **Layout** and **Header/Footer**.

In the right of the dialog on all pages you will find a preview pane which outlines how margins and header will appear when printed.

At the bottom you will find the following buttons:

Save Settings

Saves all print settings for the active sheet object.

Print Preview

Opens the **Print Preview** dialog.

Print

Starts printing and leaves the dialog.

Cancel

Exits the **Print** dialog without printing.

Help

Opens help for the dialog.

General

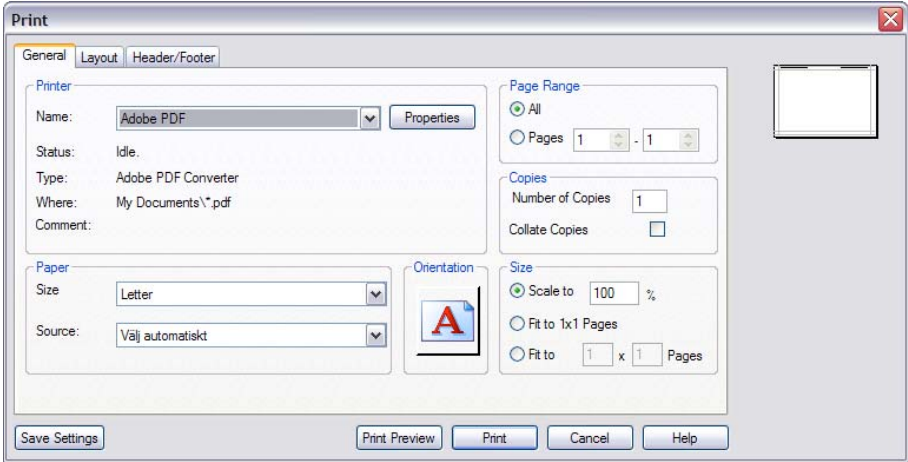


Figure 81. The General page of the Print dialog

On the **General** page of the **Print** dialog you select e.g. printer, number of copies to be printed and page range.

Printer

In the **Printer** group, you select the printer.

Name

Select a printer in the drop-down box.

Properties

Sets properties for the selected printer.

Paper

In the **Paper** group, the size and source of paper to be used are selected.

Size

Paper size can be selected in the drop-down box.

Source

Paper source can be selected in the drop-down box.

Orientation

By clicking on the orientation icon, you can choose between the print settings portrait and landscape.

Page Range

In the **Page Range** group, the number of pages to be printed is specified. If certain types of charts are selected, this group is dimmed (see **Size** below).

All

If this radio button is selected, all pages in will be printed.

Pages

Select this radion button to print the pages specified in the range to the right.

Copies

In the **Copies** group, the number of copies to be printed can be set.

Number of Copies

Here you enter the number of copies to be printed.

Collate Copies

If this check box is marked, each page will be printed together with its copies. If it is not marked, all the pages will first be printed sequentially and then copied.

Size

In the **Size** group, the size of the object to be printed can be set. If the object is a pie chart, bar chart, line chart, combo chart or scatter chart, this group will be dimmed since these charts are always fit to one page.

Scale to N%

If this option is selected, the size of the object will be scaled to a fixed percentage of its actual size.

Fit to 1x1 Pages

If this option is selected, the size of the object will be scaled to fit one page.

Fit to NxN Pages

If this option is selected, the size of the object will be scaled to fit a specified number of pages.

Layout

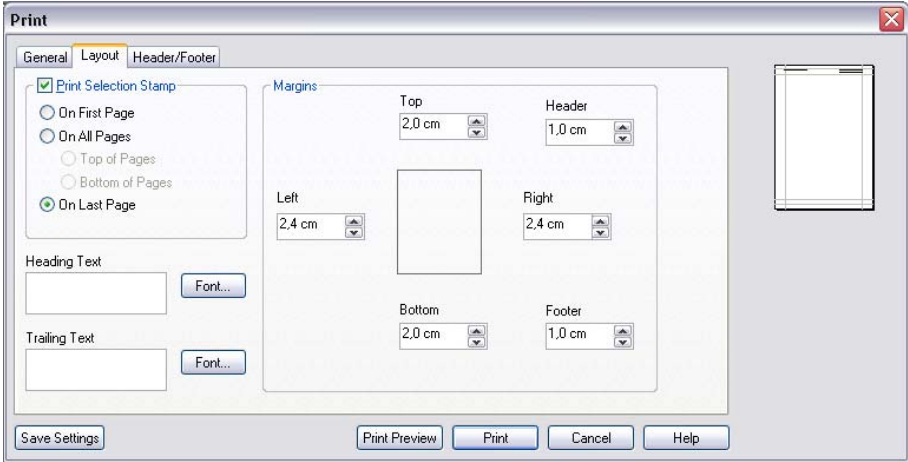


Figure 82. The Layout page in the Print dialog

On the **Layout** page of the **Print** dialog you may e.g. set the margins and heading text for the document to be printed.

Print Selection Stamp

In the **Print Selection Stamp** group you may specify that a selection stamp be printed together with the printed object. The stamp will consist of the values selected if the number of selected values falls within the value, **Max Values in Current Selections**, that is set in **User Preferences, General** page. If the number of selected values is greater, only 'x values of y' will make up the stamp. If the selection is the result of a search, the stamp will consist of the search criteria.

On First Page

Select this option to have the selection stamp printed at the top of the first page.

On All Pages

Select this option to have the selection stamp printed on every single page. There are further options for whether the selection stamp should be printed at the **Top of Pages** or at the **Bottom of Pages**.

On Last Page

Select this option to have the selection stamp printed at the bottom of the last page.

Heading Text

In the edit box you may specify a text to be printed before the printed sheet object. This text may be a calculated formula. By clicking on the ... button you open a large editor window for easier editing of longer text. The **Font** button next to the edit box allows you to select a separate font for the text.

Trailing Text

In the edit box you may specify a text to be printed after the sheet object to be printed. This text may be a calculated formula. By clicking on the ... button you open a large editor window for easier editing of longer text. The **Font** button next to the edit box allows you to select a separate font for the text.

Margins

In the **Margins** group you may change the margins around the printed object. Changes can be seen in the preview pane at the right of the **Print** dialog.

Top

Specifies the distance between the top of the paper and the upper border of the printed object.

Header

Specifies the distance between the top of the paper and the header text, if such is used. For the header text to appear this value must be smaller than **Top**.

Left

Specifies the distance between the left edge of the paper and the left border of the printed object.

Right

Specifies the distance between the right edge of the paper and the right border of the printed object.

Bottom

Specifies the distance between the bottom of the paper and the lower border of the printed object.

Footer

Specifies the distance between the bottom of the paper and the lower border of the footer text if such is used. For the footer text to appear this value must be smaller than **Bottom**.

The distance can be specified in mm, cm or ". The default unit is set on the **Design** page of **User Preferences**.

Header/Footer

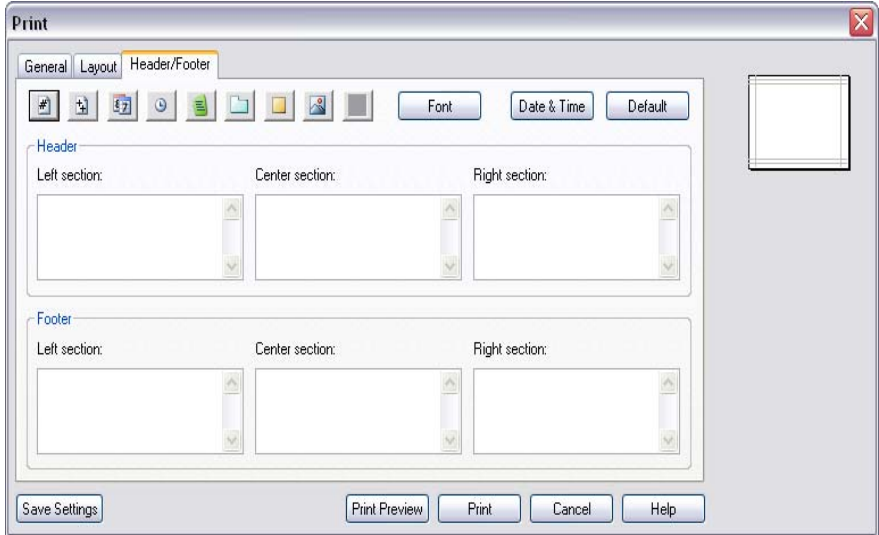


Figure 83. The Header/Footer page in the Print dialog

On the **Header/Footer** page of the **Print** dialog settings for header and footer can be set.

Header

In the **Header** group the print header can be set for the sheet object. The header can be specified in three panes.

Left Section

Text to be printed left adjusted in the header.

Center Section

Text to be printed center adjusted in the header.

Right Section

Text to be printed right adjusted in the header.

Footer

In the **Footer** group the print footer can be set for the sheet object. The footer can be specified in three panes.

Left Section

Text to be printed left adjusted in the footer.

Center Section

Text to be printed center adjusted in the footer.

Right Section

Text to be printed right adjusted in the footer.

All these text panes can contain special control codes listed below. They can also be calculated formulas (see page 467).

The following buttons can be used to insert control codes for specific system information into any of the text panes. The control codes may of course be typed directly, if you so please.

Page

Pressing this button or entering the code `&[Page]` causes the page number to be printed.



Pages

Pressing this button or entering the code `&[Pages]` causes the total number of pages to be printed.



Date

Pressing this button or entering the code `&[Date]` causes the current date to be printed. Date format can be set in the **Date & Time** dialog (see below).



Time

Pressing this button or entering the code `&[Time]` causes the current time to be printed. Time format can be set in the **Date & Time** dialog.



File

Pressing this button or entering the code `&[File]` causes the document file name to be printed.



Sheet

Pressing this button or entering the code `&[Sheet]` causes the sheet name to be printed. This option is not available when printing reports.



Title

Pressing this button or entering the code `&[Title]` causes the title of the printed object to be printed. This option is not available when printing reports.



Picture

Pressing this button will open a browser dialog where you can select a valid picture file whose content will be printed as graphics in the header or footer pane. You may also manually enter the code `&[Picture=filename]` where *filename* is the complete filename and path of the file containing the graphics. A header/footer pane can only contain one single picture and all text and other codes surrounding `&[Picture]` will be disregarded.



Font

The font of each text can be set individually. Position the cursor in the text you wish to format and use this button to set the font.

Default

Sets headers and footers to QlikView defaults.

Date & Time

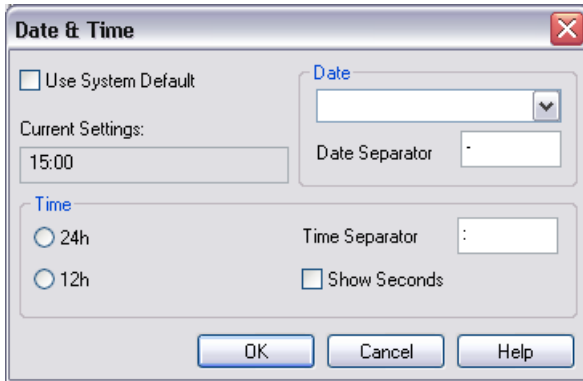


Figure 84. The Date & Time dialog

In this dialog you may set a user defined date and time format for the print job. These settings may differ from the system settings. To specify your own settings, you must first deselect the **Use System Default** check box.

Use System Default

Mark this check box to use the system settings and not the settings of this dialog.

Current Settings

A view of the current date and time format.

Date

The date format can be set here. You find the options by opening the drop-down list.

Date Separator

Here you can choose the character to be used as date separator.

Time

In the Time group settings for the time format can be set

24h

Mark this check box to show time in the 24 hour notation.

12h

Mark this check box to show time in the 12 hour notation.

Time Separator

Here you can choose the character to be used as time separator.

Show Seconds

Mark this check box to show seconds in the time format.

19.2 Print Sheet

To open this dialog box, choose **Print Sheet** from the **File** menu. The **Print Sheet** dialog box is identical to the general **Print** dialog with one exception: the **Size** group on the **General** page which is here replaced by another group, **Sheet Options**.

Sheet Options

The **Sheet Options** group contains the following settings:

This Sheet

By selecting this option only the current sheet will be printed.

All Sheets

By selecting this option all sheets in the document will be printed.

Draw Background

Mark this check box to include the sheet background (wallpaper) when printing.

19.3 Print Preview

By choosing the command **Print Preview** from the **File** menu or clicking on the **Print Preview** button in the **Print** dialog, you open a window showing a detailed print preview of the current object.

Print

Transfers control the **Print** dialog, from which you can print the active object.

#pages drop-down

This drop-down menu offers a quick way to change which pages should be displayed in the preview.

Next

If the preview contains several pages, you click this button to view the next page.

Previous

If the preview contains several pages, you click this button to view the previously shown page.

Add Page

Extends the preview to include another page if the active object does not fit in one page.

Remove Page

Removes one of the preview pages.

Close

Closes the dialog.

19.4 Copy Mode

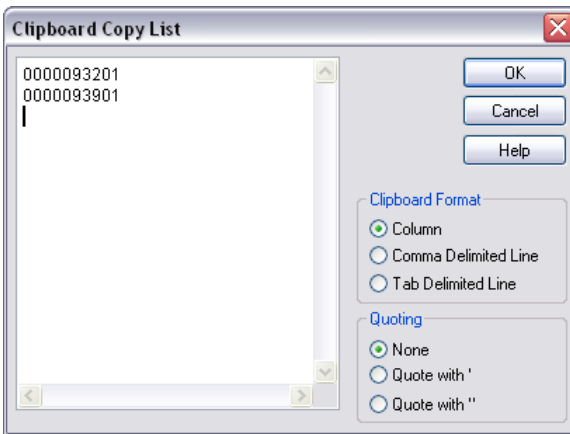


Figure 85. The Clipboard Copy List dialog

When **Copy Mode** in the **Edit** menu is chosen, the **Clipboard Copy List** dialog (Figure 85) opens. This dialog box simplifies copying to clipboard. As long as it is open, QlikView will be in copy mode and everything clicked will automatically be copied to the clipboard copy list.

No new logical selections can be made while in copy mode.

Clipboard Format

In this group, the format of the copy list is set. **Column**, **Comma Delimited Line** and **Tab Delimited Line** are the available options.

Quoting

In this group the quoting for the selected elements can be set.

None

Leaves the elements unquoted.

Quote with '

Encloses all elements with single quotes. This is useful if the elements copied are to be pasted into the script as field values.

Quote with "

Encloses all elements with double quotes. This is useful if the elements copied are to be pasted into the script as field names or into a Visual Basic script, e.g. a QlikView macro.

OK

Closes the **Clipboard Copy List** dialog and transfers its content to the Windows clipboard.

Cancel

Closes the **Clipboard Copy List** dialog without transferring its content to the Windows clipboard.

19.5 Export to File

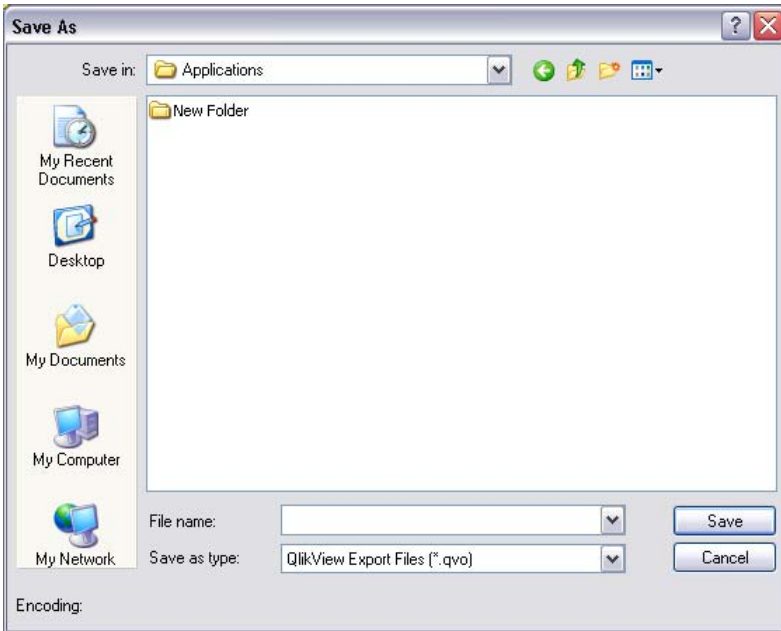


Figure 86. The Save As dialog

The **Save As** dialog box appears when you choose **Export...** from the context menu of the table box, pivot table or straight table.

In this dialog you can define the file to which the data should be exported (export file).

Save in:

Browse to find the directory containing your file.

File name

The name of the file selected will appear in this list. It is also possible to define a new file by entering a name.

Save as type

Here you can choose an appropriate format for your export file by selecting one of the options in the drop-down list. You can choose between Comma Delimited, Semicolon Delimited, Tab Delimited, Hypertext (HTML), XML and Excel (xls). The default format is *.qvo (QlikViewOutput), a tab separated file.

20 LAYOUT THEMES

20.1 About QlikView Layout Themes

Theme Basics

A QlikView layout theme is a set of formatting properties that can be applied to a whole QlikView layout or to parts thereof. Theme files are coded in xml and normally kept in a special QlikView theme folder under Windows' Application Data folder for the active user.

Themes can significantly reduce time and effort needed to make a QlikView layout good-looking and consistent. They can also be used to enforce a certain graphical style within a company etc.

Theme Topology

A QlikView theme file consists of individual sections for each type of layout object (document, sheet and all different types of sheet objects). Each of these sections has three sub-sections, one for object type specific properties, one for border/caption properties and one for printer setting properties (only for printable object types). Each section and sub-section can be created or replaced separately using the **Theme Maker Wizard**, without affecting other sections in the theme file.

A theme section is created as follows:

- 1 Format a specific object according to your wishes
- 2 Use the **Theme Maker Wizard** to extract selected properties from that object and insert them into a theme file

General Purpose Themes

General purpose themes are themes which have defined sections for all different object types. There are a number of such themes provided with the QlikView installation package.

If you want to create such a theme yourself you must:

- 1 Format **Document Properties** to your liking.
- 2 Format the **Sheet Properties** of one sheet to your liking.

-
- 3 Create (if necessary) and format one sheet object of each type to your liking. Caption/Border properties typically only need to be formatted once for sheet object types which normally use captions (list boxes, charts etc.) and once those which normally do not use captions (text objects, buttons etc.).
 - 4 Run the **Theme Maker Wizard** creating a new theme and insert properties from the first of the formatted entities listed above (in what order doesn't matter).
 - 5 Run the **Theme Maker Wizard** repeatedly, once for every remaining formatted entity listed above.

Specific Purpose Themes

It is possible to create themes for specific uses. Depending on the purpose these may only need to define a small sub-set of the available theme sections and sub-sections. As an example you may define a theme which sets only chart objects to a given size and position on the sheet. In order to do that you need to define a theme with one single property from the caption/border group only for charts, i.e. one single sub-section.

20.2 Applying Themes in the Layout

QlikView layout themes can be applied either manually at any given time or automatically every time a new layout object is created. Themes can be applied to

- individual sheet objects
- a group of sheet objects
- a sheet
- the whole document

Applying a Theme to Individual Objects

Do as follows:

- 1 Activate the sheet object by clicking it.
- 2 Open the **Properties** dialog for the sheet object from the context menu.
- 3 Go to the **Layout** page.
- 4 Click the **Apply Theme...** button.
- 5 Select a theme in the browser dialog that opens.
- 6 Click **OK**.

All properties in the theme which are applicable to the selected sheet object will now be applied. If you want to apply a theme to a group of sheet objects, you must first make them all active by means of SHIFT+clicking or by doing a lasso selection.

Applying a Theme to a Sheet

Do as follows:

- 1 Activate the sheet by clicking its tab.
- 2 Open the **Sheet Properties** dialog from the **Settings** menu.
- 3 Go to the **General** page.
- 4 Click the **Apply Theme...** button.
- 5 Select a theme in the browser dialog that opens.
- 6 Click **OK**.

All properties in the theme which are applicable to the document will now be applied. Additionally the theme will also be applied to all sheets and all sheet objects in the document.

Applying a theme to an entire document

Do as follows:

- 1 Open the document or activate it.
- 2 Open the **Document Properties** dialog from the **Settings** menu.
- 3 Go to the **Layout** page.
- 4 Click the **Apply Theme...** button.
- 5 Select a theme in the browser dialog that opens.
- 6 Click **OK**.

All properties in the theme which are applicable to the document will now be applied. Additionally the theme will also be applied to all sheets and all sheet objects in the document.

20.3 Making Themes with the Theme Maker Wizard

You start the Theme Maker Wizard by choosing **Theme Maker Wizard** from the **Layout** menu.



Figure 87. The Start page of the Theme Maker Wizard

The first time you start the wizard you will be met by a start page outlining the purpose of the wizard and the basic steps involved. If you want to skip the start page when you use the wizard in the future, mark the **Don't show this page again** check box.

Click **Next** to continue.

Step 1 - Select Theme File

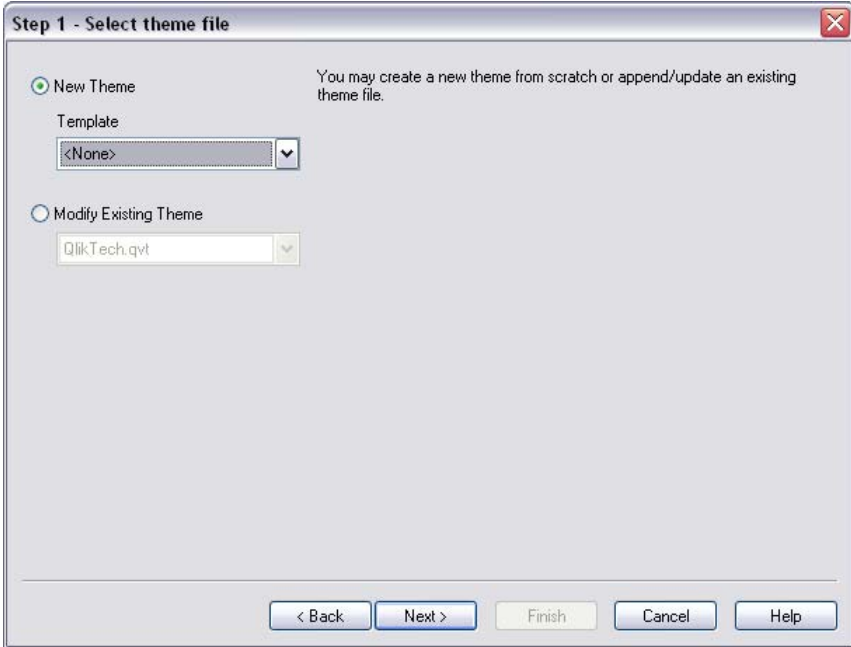


Figure 88. The first page of the Theme Maker Wizard

You can choose between creating a new theme from scratch, creating a new theme based on an existing one or modifying an existing theme.

New Theme

Choose this option if you want to create a new theme.

Template

If you want to base your new theme on an existing one, choose the base theme in this drop-down. The drop-down will list all existing themes in your default QlikView theme folder. At the bottom of the list you will find a **Browse...** option for browsing for theme files in other locations.

Modify Existing Theme

Choose this option if you want to modify an existing theme. Select a theme in the drop-down. The drop-down will list all existing themes in your default QlikView theme folder. At the bottom of the list you will find a **Browse...** option for browsing for theme files in other locations.

Click **Next** to continue. If you have chosen to create a new theme, a Save As dialog will appear.

Step 2 - Source Selection

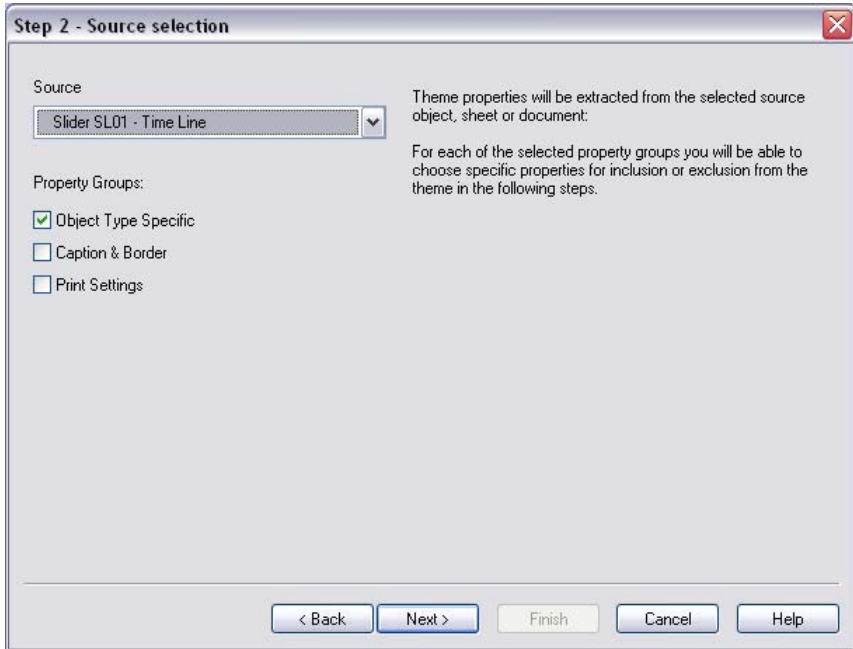


Figure 89. The second page of the Theme Maker Wizard

When building a theme you take one or more groups of formatting properties from an existing object. The source object can be any sheet object, any sheet or even the document itself.

Source

Select the source object from the list in the drop-down. The drop-down list contains a list of all available objects in the document. The active object will be pre-selected.

Property Groups

There are three main groups of formatting properties that can be extracted from a layout object and inserted into a theme. Mark one or more of the three check boxes below for extraction from the source object to the theme:

Object Type Specific

Mark this check box to extract object type specific properties from the source object for inclusion in the theme.

Object type specific properties are those that only exist in a given object type, e.g. charts. This type of properties can only be copied to objects of the same type as the source object type.

Caption and Border

Mark this check box to extract caption and border properties from the source object for inclusion in the theme. This type of properties can be copied to other object types than the source object type.

Printer Settings

Mark this check box to extract printer settings properties from the source object for inclusion in the theme. This type of properties can be copied to other object types than the source object type.

For each of the selections above you will be able to choose specific properties for inclusion in or exclusion from the theme in the following wizard steps.

Click **Next** to continue.

Step 3 - Select Specific Properties

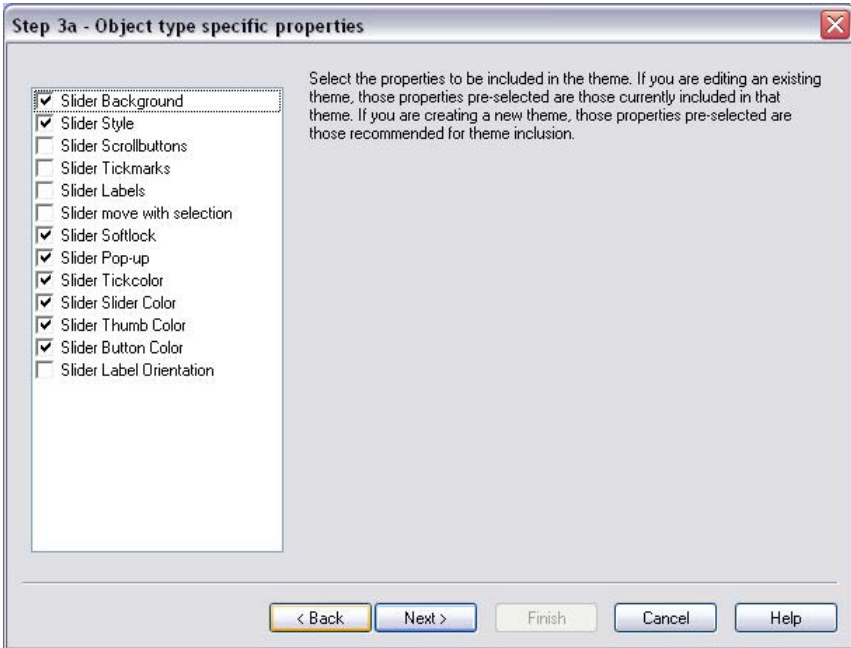


Figure 90. The third page of the Theme Maker Wizard as it could appear for slider specific properties.

In step 3 you will make more detailed selections of properties to be extracted from the source object and included in the theme. This wizard step will be repeated for each of the three main property groups selected in step 2.

Each item in the list indicates a single property or a group of properties, which can be included in or excluded from the theme. Mark those items that you want to include.

If you are modifying an existing theme, those items marked as you enter this step are those currently included in the theme. By changing selections you will overwrite all previous settings in the theme.

If you are creating a new theme from scratch, the items that are marked as you enter this step are those which are typically suitable for inclusion in a general purpose theme.

Click **Next** to continue.

Step 4 - Insertion of Properties in a Theme

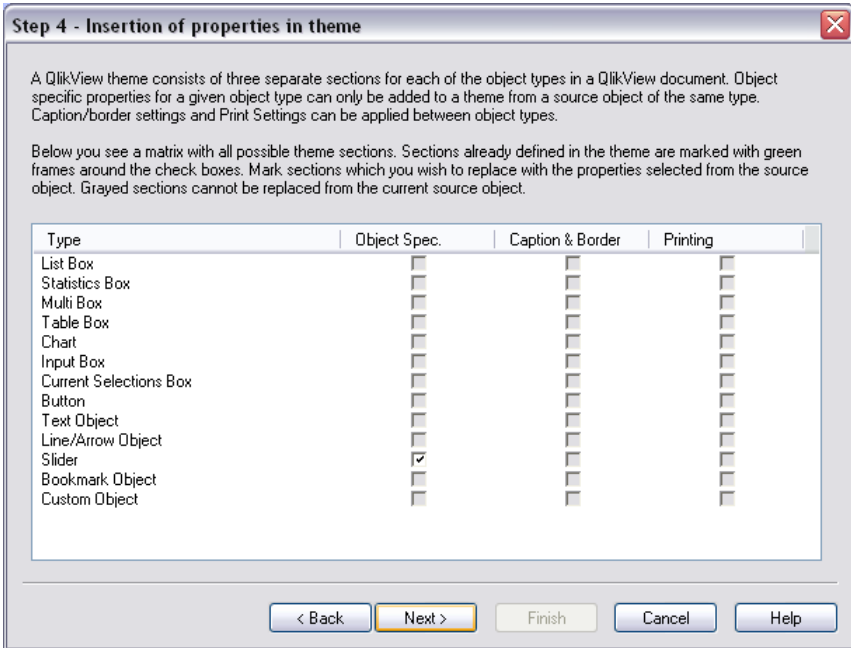


Figure 91. The fourth page of the Theme Maker Wizard

In the fourth step of the wizard you decide which sections and sub-sections of the theme should be written to the theme with the extracted object properties. There are three columns with check boxes, each check box representing one sub-section of the theme. Based on your selections in step 2 and 3 of the wizard only some of the check boxes will be available for selection, the remainder being grayed out.

Check boxes surrounded by green frames indicate sub-sections that are currently defined in the theme (only applicable when modifying an existing theme).

The columns with check boxes correspond to the three check boxes in step 2. It is only possible to make selections in a column if the corresponding check box was selected in step 2, followed by appropriate selections in step 3.

The object type specific sub-section can only be selected for the type of source object selected in step 2. Caption/border settings and printer settings can be set across object types.

Click **Next** to continue.

Step 5 - Save Theme

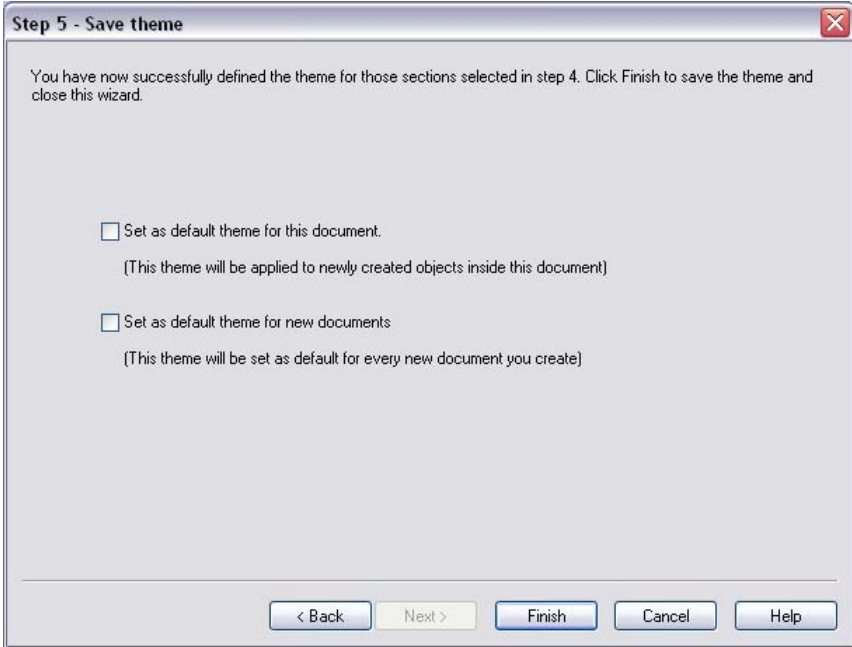


Figure 92. The fifth page of the Theme Maker Wizard

You have now concluded the process of creating or modifying a theme. In case you would like to use the saved theme as defaults for new documents or objects there are two options available:

Set as Default Theme for this Document

Mark this check box to use this theme as default theme in the current document. This means that it will be applied to all created sheets and sheet objects in the document. The selected theme must be accessible from disc at all times in order to be used. It is also important that the theme used is defined for all types of objects that may occur in a QlikView document. Default themes can be set at all times from the **Presentation** page of the **Document Properties** dialog.

Set as Default Theme for New Document

Mark this check box to use this theme as default theme for new documents. This means that it will be set as default theme in new documents. The selected theme must be accessible from disc at all times in order to be used. It is also important that the theme used is

defined for all types of objects that may occur in a QlikView document. Default themes new documents can be set at all times from the **Design** page of the **User Preferences** dialog.

Click **Finish** to save the theme and return to the layout.



21 OBJECTS CLIENT PAGE GENERATOR

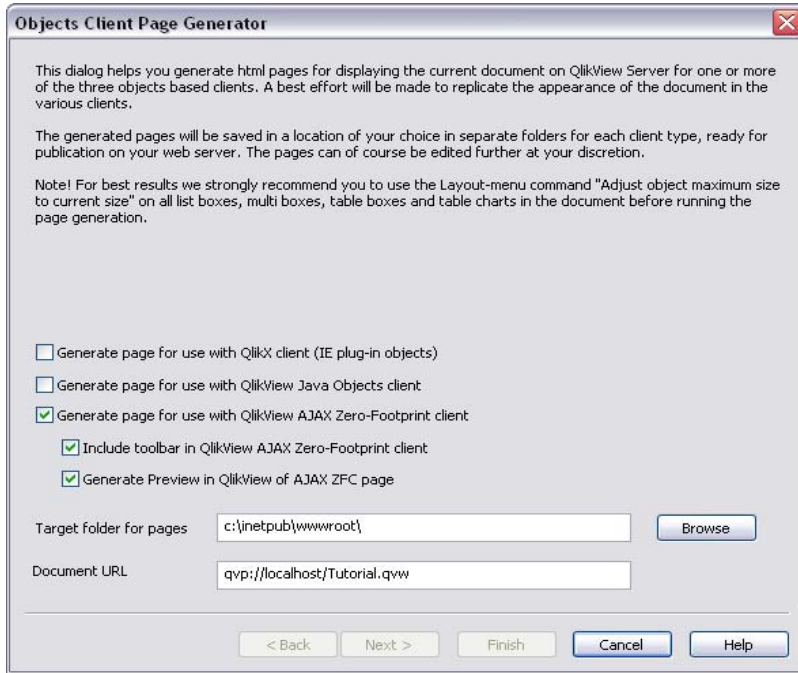


Figure 93. The Objects Client Page Generator dialog

This dialog helps you generate html pages for displaying the current document on QlikView Server for one or more of the three objects based clients. A best effort will be made to replicate the appearance of the document in the various clients. The generated pages will be saved in a location of your choice in separate folders for each client type, ready for publication on your web server. The pages can of course be edited further at your discretion, using a third-party html editor. For the best possible result, the Server generating the pages should be the same version as the clients using the them.

As of version 9, pages for all clients are generated on demand by the QlikView Server.

The following settings can be made:

Generate page for use with QlikX client (IE plug-in objects)

If this check box is marked, a page will be generated for the object version of the QlikView IE plug-in (a.k.a. QlikX).

Generate page for use with QlikView Java Objects client

If this check box is marked, a page will be generated for the object version of the QlikView Java client (a.k.a. QlikView Java Objects).

Generate page for use with QlikView AJAX Zero-Footprint client

If this check box is marked, a page will be generated for the object version of the QlikView AJAX Zero-Footprint client.

Include Toolbar in QlikView AJAX Zero-Footprint client

This option is checked by default, and will include a basic navigation toolbar in the generated pages.

Generate Preview in QlikView of AJAX ZFC page

If this check box is marked, a page will be generated for the object version of the QlikView IE plug-in (a.k.a. QlikX).

Note The AJAX ZFC pages must be generated on the same version of QlikView as the Server version!

Target folder for pages

Specifies the folder where the generated pages will be saved. Each page variant (for different client types) will be saved in a separate sub-folder in the specified location.

Document URL

Specifies the target document URL on the QlikView Server where the pages are to be run. This setting is necessary for the AJAX Zero Footprint Client, the Java Objects and QlikX pages to work properly.

PART II: NUMBER FORMATS

- **Data Types in QlikView**
- **Number Format Dialogs**
- **Format Codes**



22 DATA TYPES IN QLIKVIEW

QlikView can handle text strings, numbers, dates, times, time stamps and currencies correctly. They can be sorted, displayed in a number of different formats and they can be used in calculations. This means e.g. that dates, times and time stamps can be added to or subtracted from each other.

22.1 Data Representation inside QlikView

In order to understand data interpretation and number formatting in QlikView, it is necessary to know how data is stored internally by the program. All of the data loaded into QlikView is available in two representations, as a string and as a number.

- 1 The string representation is always available and is what is shown in the list boxes and the other sheet objects. Formatting of data in list boxes (number format) only affects the string representation.
- 2 The number representation is only available when the data can be interpreted as a valid number. The number representation is used for all numeric calculations and for numeric sorting.

If several data items read into one field have the same number representation, they will all be treated as the same value and will all share the first string representation encountered. Example: The numbers 1.0, 1 and 1.000 read in that order will all have the number representation 1 and the initial string representation 1.0.

22.2 Number Interpretation

Data With Type Information

Fields containing numbers with a defined data type in a database loaded via ODBC will be handled by QlikView according to their respective formats. Their string representation will be the number with an appropriate formatting applied.

QlikView will remember the original number format of the field even if the number format is changed in the number format dialogs of the application. The original format can always be restored by clicking on the **Default from input** button in the number format dialogs.

The QlikView default formats for the different data types are:

- integer, floating point numbers the default setting for number

-
- currency: the default setting for currency
 - time, date, timestamp: ISO standard formatting

The default settings for number and currency are defined via the script number interpretation variables (see page 347 in Book I) or the operating system settings (**Control Panel**).

Example:

Original data:

type = numeric (1 decimal)
1.0
2.5
30.0

Data read into QlikView (via ODBC)

Original format info

fix '0.0'

Format info

fix '0.0'

String value (list box)

1.0
2.5
30.0

Number value (calc)

1
2.5
30

sum(A) = 33.5

Apply new number format **currency '\$0.00'**

money '\$0.00'

\$1.00
\$2.50
\$30.00

1
2.5
30

sum(A) = 33.5

Apply new number format **'Default from input'**

fix '0.0'

1.0
2.5
30.0

1
2.5
30

sum(A) = 33.5

Data Without Type Information

For data without specific formatting information from the source (e.g. data from text files or ODBC data with a general format) the situation becomes more complicated. The final result will depend on at least five different factors:

- 1 The way data is written in the source database

-
- 2 The operating system settings for number, time, date etc. (**Control Panel**)
 - 3 The use of optional number interpreting variables in the script
 - 4 The use of optional interpretation functions in the script
 - 5 The use of optional formatting functions in the script
 - 6 The number formatting controls in the document

QlikView tries to interpret input data as a number, date, time etc. As long as the system default settings are used in the data, the interpretation and the display formatting is done automatically by QlikView, and the user does not need to alter the script or any setting in QlikView. There is an easy way to find out if the input data has been correctly interpreted: numeric values are right-aligned in list boxes, whereas text strings are left-aligned.

By default, the following scheme is used until a complete match is found. (The default format is the format such as the decimal separator, the order between year, month and day, etc. specified in the operating system, i.e. in the **Control Panel**, or in some cases via the special number interpretation variables in the script, see page 347 in Book I.)

QlikView will interpret the data as

- 1 a number according to the default format for numbers.
- 2 a date according to the default format for date.
- 3 a time stamp according to the default format for time and date.
- 4 a time according to the default format for time.
- 5 a date according to the following format: yyyy-MM-dd.
- 6 a time stamp according to the following format: YYYY-MM-DD hh:mm[:ss[.fff]].
- 7 a time according to the following format: hh:mm[:ss[.fff]].
- 8 money according to the default format for currency.
- 9 a number with '.' as decimal separator and ',' as thousands separator, provided that neither the decimal separator nor the thousand separator is set to ','.
- 10 a number with ',' as decimal separator and '.' as thousand separator, provided that neither the decimal separator nor the thousand separator is set to '.'.

- 11 a text string. This last test never fails: if it is possible to read the data, it is always possible to interpret it as a string.

When loading numbers from text files some interpretation problems may occur, e.g. an incorrect thousand separator or decimal separator may cause QlikView to interpret the number incorrectly. The first thing one should do is to check that the number interpretation variables in the script (see page 347 in Book I) are correctly defined and that the system settings in the **Control Panel** are correct.

Once QlikView has interpreted data as a date or time, it is possible to change to another date or time format in the properties dialog for the sheet object.

Since there is no predefined format for the data, different records may of course contain differently formatted data in the same field. You may e.g. find valid dates, integers and text in one field. The data will therefore not be formatted, but shown in its original form in list boxes etc.

When you first enter a number format dialog for such a field, the format will be set as **Mixed**. As soon as you change the number formatting of the field, it will be impossible for QlikView to return to the original formatting of the different field values, unless the script is re-executed with the **Survive Reload** check box deselected. The **Default from input** button in the number format dialogs is thus unavailable for this kind of fields once the number format has been changed.

Example:

Original data:

1
2.4
3.1415
abc123

Data read into QlikView (from text file)

Original format info

not available

Format info

mixed

String value (list box)

1
2.4
3.1415
abc123

Number value (calc)

1
2.4
3.1415
-
sum(A) = 6.5415

Apply number format **integer '0'**

integer '0'

1
2
3
abc123

1
2.4
3.1415
-
sum(A) = 6.5415

Apply number format **money '\$ #.00'**

money '\$ #.00'

\$1.00
\$2.40
\$3.14
abc123

1
2.4
3.1415
-
sum(A) = 6.5415

22.3 Dates and Times

QlikView stores each date, time and time stamp found in data as a date serial number. The date serial number is used for dates, times and time stamps and in arithmetic cal-

culations based on date and time entities. Dates and times can thus be added and subtracted, intervals can be compared, etc.

The date serial number is the (real valued) number of days passed since Dec 30, 1899, i.e. the QlikView format is identical to the 1900 date system used by Microsoft Excel for Windows, Lotus 1-2-3, Corel Quattro Pro and other programs, in the range between March 1, 1900 and Feb 28, 2100. For example, 33857 corresponds to Sept 10, 1992. Outside this range, QlikView uses the same date system extended to the Gregorian calendar, which is now the standard calendar in the Western World.

The serial number for times is a number between 0 and 1. The serial number 0.00000 corresponds to 00:00:00, whereas 0.99999 corresponds to 23:59:59. Mixed numbers indicate the date and time: the serial number 2.5 represents January 1, 1900 at 12:00 noon.

The data is however displayed according to the format of the string. By default, the settings made in the **Control Panel** are used. It is also possible to set the format of the data by using the number interpretation variables in the script (see page 347 in Book I) or with the help of a formatting function, see page 453 in Book I. Finally it is also possible to reformat the data in the properties sheet of the sheet object.

Examples:

1997-08-06	is stored as	35648
09:00	is stored as	0.375
1997-08-06 09:00	is stored as	35648.375

or vice versa

35648	with number format 'D/M/YY' is shown as	6/8/97
0.375	with number format 'hh.mm' is shown as	09.00

As mentioned earlier, QlikView will follow a set of rules to try to interpret dates, times and other data types. The final result however will be affected by a number of factors as described above.

Examples:

These examples assume the following default settings:

number decimal separator:	.
short date format:	YY-MM-DD
time format:	hh:mm

The following data is read into QlikView as the field A:

Original data:

0.375
33857
97-08-06
970806
8/6/97

Inside QlikView:

A1) QlikView interpretation (without the special interpretation function in the script)

String value (list box)	Number value
0.375	0.375
33857	33857
97-08-06	35648
970806	970806
8/6/97	-

A2) With number format **date 'YYYY-MM-DD'**

String value (list box)	Number value
1899-12-30	0.375
1992-09-10	33857
1997-08-06	35648
4557-12-21	970806
8/6/97	-

A3) With number format **date 'MM/DD/YYYY'**

String value (list box)	Number value
12/30/1899	0.375
09/10/1992	33857
08/06/1997	35648
12/21/4557	970806
8/6/97	-

A4) With number format **time 'hh:mm'**

String value (list box)	Number value
09:00	0.375
00:00	33857
00:00	35648
00:00	970806
8/6/97	-

A5) With number format **fix '# ##0.00'**

String value (list box)	Number value
0.38	0.375
33 857.00	33857
35 648.00	35648
970 806.00	970806
8/6/97	-

B1) QlikView interpretation with the special interpretation function **date#(A, 'M/D/YY')**

String value (list box)	Number value
0.375	-
33857	-
97-08-06	-
970806	-
8/6/97	35648

B2) With number format **date 'YYYY-MM-DD'**

String value (list box)	Number value
0.375	-
33857	-
97-08-06	-
970806	-
1997-08-06	35648

B3) With number format **date 'MM/DD/YYYY'**

String value (list box)	Number value
0.375	-
33857	-
97-08-06	-
970806	-
08/06/1997	35648

B4) With number format **time 'hh:mm'**

String value (list box)	Number value
0.375	-
33857	-
97-08-06	-
970806	-
00:00	35648

B5) With number format **fix '# ##0.00'**

String value (list box)	Number value
0.375	-
33857	-
97-08-06	-
970806	-
35 648.00	35648

23 NUMBER FORMAT DIALOGS

The format for dates, times and numbers for any field can be set either in the script (see page 453) or in the **Number** page of the properties dialogs. Here you can choose a format different from the one used by the system. Each field can only have one number format at any given time within a sheet object. Changes made to a field in the **Number** page in the **Document Properties** dialog will affect the formatting of the field in all sheet objects. It is however possible to use a separate number format in an individual sheet object. Mark the check box **Override Document Settings** on the **Number** page in the sheet object's properties dialog and specify a number format in the group control below.

23.1 Document Properties

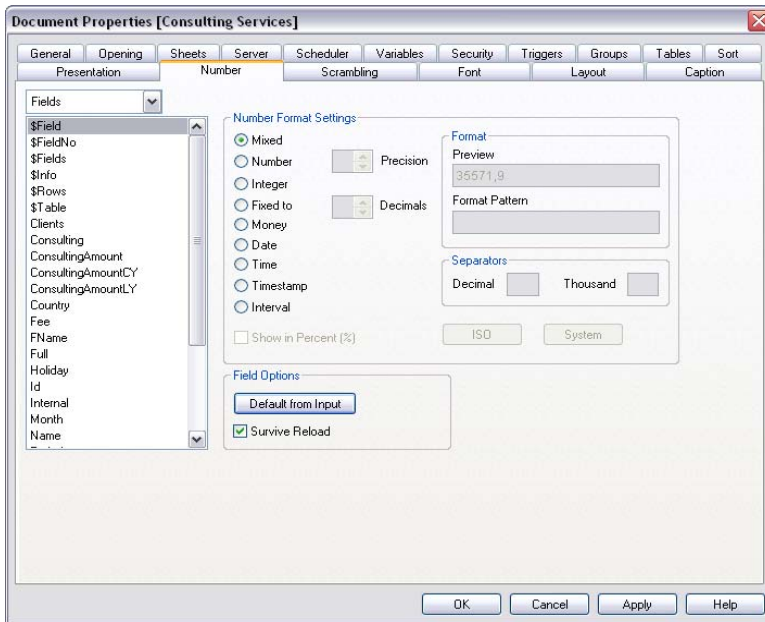


Figure 94. The Number page in the Document Properties dialog

Fields

A list of the fields or variables available for formatting. Mark the field(s) or variable(s) for which you want to set the number format. Several field(s) or variable(s) can be marked simultaneously.

Variables

A list of the variables available for formatting. Mark the variable or variables for which you want to set the number format. Several variables can be marked simultaneously.

Field Options

Default from input

Resets the format (if possible) to the input format as interpreted by the script. This option is only available for fields loaded with specific type information.

Survive Reload

If this control is checked, the set number format will be kept also after a reexecution of the script. If not, the format will be reset to its original settings at the reexecution.

Mixed

Field values of different types will be shown with different formatting. Initially, all fields without specific type information are set to **Mixed**.

Number

Numeric field values will be shown as numbers with the number of significant digits as set in the **Precision** box.

Integer

Numeric field values will be shown as integer numbers.

Fixed to

Numeric field values will be shown as numbers with a fix number of decimals as set in the **Decimals** box.

Money

Numeric field values will be shown in the money format.

Date

Numeric field values will be shown as dates.

Time

Numeric field values will be shown as times.

Timestamp

Numeric field values will be shown as time stamps.

Interval

Numeric field values will be shown as intervals.

Note If you change from the **Mixed** format to some other number format, you may lose the original formatting information, i.e. QlikView will then not be able to distinguish between numbers and dates.

Show in Percent (%)

Click this button to toggle formatting as percentage instead of absolute numbers on and off. Percentage formatting is only available for **Number**, **Integer** and **Fix**.

Format

The format code that further specifies the display format of the field. See also page 307.

Preview

A preview based on the format code specified above.

Separators

In this group it is possible to set a decimal separator and a thousand separator different from the system separators.

ISO

Sets the format to the ISO standard. Only valid for date, time, time stamp and interval.

System

Sets the format to the system settings.

23.2 List Box, Multi Box, Table Box and Input Box

The list box, multi box and table box properties dialogs also contain **Number** pages. Each field has a default number format which can be set on the **Number** page of the **Document Properties** dialog as described in the previous section. Normally this is the number format used when displaying field values in a list box, multi box or table box. It is however possible to use a separate number format in an individual sheet object. To do this, mark this check box **Override Document Settings** and specify a number format in the group control below.

The **Number** pages of list boxes, multi boxes and table boxes otherwise differ from the **Number** page of the **Document Properties** dialog only regarding the **Fields** list. Since a list box contains only one field, the **Fields** list is not available in the **Number** page of the **List Box Properties** dialog. As for the **Fields** list in the multi box or table box properties **Number** page, it contains only the fields currently shown in the concerned multi box or table box.

The **Input Box Properties** dialog contains a **Number** page, that in large resembles to the dialogs described above. The list however only contains the variables shown in the input box.

23.3 Chart

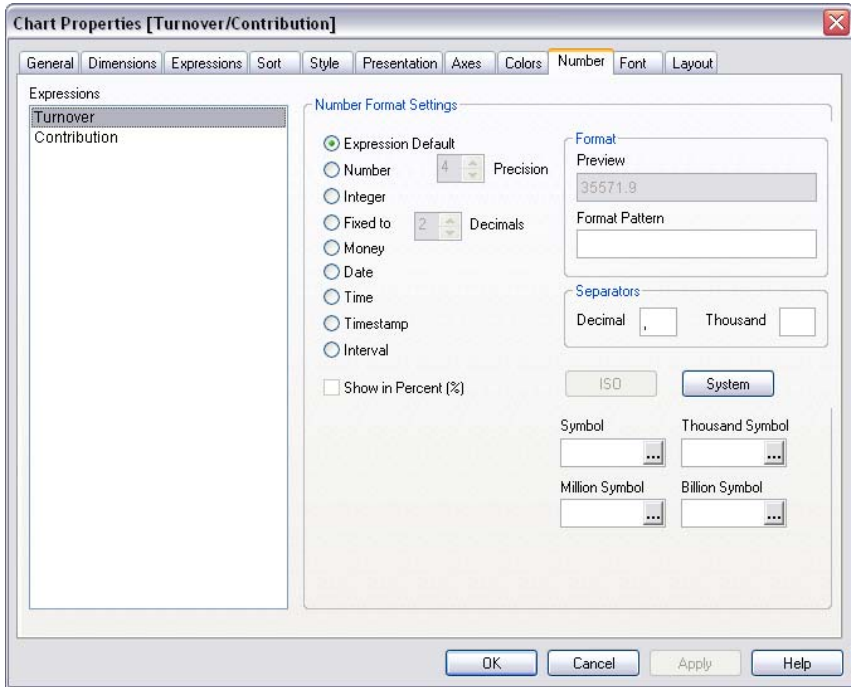


Figure 95. The Number page in the Chart Properties dialog

The Number page in the Chart Properties dialog differs from the others in that the number format is only available for the expressions. Setting the number format for an expression does not affect the number format setting for the fields. Each chart can have its own number formats.

If you want to change the number format for the dimensions in e.g. a pivot table, this must be done in the **Document Properties Number** page.

However, if the x-axis is set to **Continuous** on the **Axes** page, the x-axis dimension is treated as an expression.

Expressions

A list of the expressions in the chart. Mark the expression for which you want to set the number format. Several expressions can be marked simultaneously. If the x-axis is set to **Continuous** on the **Axes** page, the x-axis dimension is also treated as an expression.

Expression Default

The inherent number format of the chart expression.

Symbol

The symbol to use as unit.

1 000 Symbol

The symbol to use for one thousand units. The symbol may be defined as a calculated formula (see page 467 in Book III) for dynamic update.

1 000 000 Symbol

The symbol to use for one million units. The symbol may be defined as a calculated formula (see page 467 in Book III) for dynamic update.

1 000 000 000 Symbol

The symbol to use for one billion units. The symbol may be defined as a calculated formula (see page 467 in Book III) for dynamic update.



24 FORMAT CODES

In the number format controls described above and in several of the interpretation and formatting functions it is possible to set the format for numbers and dates by using a *format code*. This describes how to format a number, date, time or time stamp.

24.1 Numbers

- To denote a specific number of digits, use the symbol "0" for each digit.
- To denote a possible digit, use the symbol "#". If the format contains only #'s to the left of the decimal point, numbers less than 1 begin with a decimal point.
- To mark the position of the thousand separator or the decimal separator, use the thousand separator and the decimal separator.

Examples:

- # ##0 describes the number as an integer with a thousand separator.
- ###0 describes the number as an integer without a thousand separator.
- 0000 describes the number as an integer with at least four digits. E.g. the number 123 will be shown as 0123.
- 0.000 describes the number with three decimals.
- 0.0## describes the number with at least 1 decimal and at most three decimals.

The format code is used for defining the positions of the separators. It is not possible to set the separator in the format code. Use the respective control (in dialogs) or parameter (in script functions) for this.

It is possible to use the thousand separator to group digits by any number of positions. E.g a format string of "0000-0000-0000" (thousand separator="-") could be used to display a ten digit part number as 12-4567-8912.

Special number formats

QlikView can interpret and format numbers in any radix between 2 and 36 including binary, octal and hexadecimal. It can also handle roman formats.

To indicate binary format the format code should start with (bin) or (BIN).

To indicate octal format the format code should start with (oct) or (OCT).

To indicate binary format the format code should start with (hex) or (HEX). If the capitalized version is used A-F will be used for formatting (e.g. 14FA).



The non-capitalized version will result in formatting with a-f (e.g. 14fa). Interpretation will work for both variants regardless of the capitalization of the format code.

The use of (dec) or (DEC) to indicate decimal format is permitted but unnecessary.

To indicate a format in any radix between 2 and 36 the format code should start with (rxx) or (Rxx) where xx is the two-digit number denoting the radix to be used. If the capitalized R is used letters in radices above 10 will be capitalized when QlikView is formatting (e.g. 14FA). The non-capitalized r will result in formatting with non-capital letters (e.g. 14fa). Interpretation will work for both variants regardless of the capitalization of the format code. Note that (r02) is the equivalent of (bin), (R16) is the equivalent of (HEX) and so on.

To indicate roman numbers the format code should start with (rom) or (ROM). If the capitalized version is used capital letters will be used for formatting (e.g. MMXVI). The non-capitalized version will result in formatting with lower cap letters (mmxvi). Interpretation will work for both variants regardless of the capitalization of the format code. Roman numbers are generalized with minus sign for negative numbers and 0 for zero. Decimals are ignored with roman formatting.

Examples:

num(199, '(bin)')	returns 11000111
num(199, '(oct)')	returns 307
num(199, '(hex)')	returns c7
num(199, '(HEX)')	returns C7
num(199, '(r02)')	returns 11000111
num(199, '(r16)')	returns c7
num(199, '(R16)')	returns C7
num(199, '(R36)')	returns 5J
num(199, '(rom)')	returns cxcix
num(199, '(ROM)')	returns CXCIX

24.2 Dates

- To describe the day, use the symbol "D" for each digit.

- To describe the month number, use the symbol "M" or "MM" for one or two digits. "MMM" denotes short month name in letters as defined by the operating system or by the override system variable MonthNames in the script. "MMMM" denotes long month name in letters as defined by the operating system or by the override system variable LongMonthNames in the script.
- To describe the year, use the symbol "Y" for each digit.
- To describe the weekday, use the symbol "W". One W will return the number of the day (e.g. 0 for Monday) as a single digit. "WW" will return the number with two digits (e.g. 02 for Wednesday). "WWW" will show the short version of the weekday name (e.g. Mon) as defined by the operating system or by the override system variable DayName in the script. "WWWW" will show the long version of the weekday name (e.g. Monday) as defined by the operating system or by the override system variable LongDayName in the script
- Arbitrary separators can be used.

Examples:

YY-MM-DD	describes the date as 01-03-31.
YYYY-MM-DD	describes the date as 2001-03-31.
YYYY-MMM-DD	describes the date as 2001-Mar-31.
31 MMMM YYYY	describes the date as 31 March 2001.
M/D/Y	describes the date as 3/31/01.
W YY-MM-DD	describes the date as 6 01-03-31.
WWW YY-MM-DD	describes the date as Sat 01-03-31.
WWWW YY-MM-DD	describes the date as Saturday 01-03-31.

24.3 Times

- To describe the hours, use the symbol "h" for each digit.
- To describe the minutes, use the symbol "m" for each digit.
- To describe the seconds, use the symbol "s" for each digit.
- To describe the fractions of a second, use the symbol "f" for each digit.
- To describe the time in AM/PM format, use the symbol "tt" after the time.
- Arbitrary separators can be used.

Examples:

hh:mm	describes the time as 18:30
hh.mm.ss.ff	describes the time as 18.30.00.00

hh:mm:tt describes the time as 06:30:pm

24.4 Time Stamps

The same notation as that of dates and times above is used.

Examples:

YY-MM-DD hh:mm describes the timestamp as 97-03-31 18:30

M/D/Y hh.mm.ss.ffff describes the timestamp as 3/31/97

18.30.00.0000

PART III: MACROS AND AUTOMATION

- **Edit Module Dialog Box**
- **About the VBScript Language**
- **QlikView Automation Interface**



25 USING AUTOMATION AND MACROS WITH QLIKVIEW

25.1 The QlikView Automation Interface

QlikView is equipped with an Automation interface (Automation was previously known as OLE Automation). This interface allows an external program or internal macro to access and control the QlikView application.

The complete specification of the QlikView Automation interface can be found in the folder Documentation of your QlikView application.

The Automation interface is an integral part of QlikView and you do not have to perform any special tasks to activate it.

25.2 How Automation and Macros Can Control QlikView

External Control of QlikView

QlikView objects are accessible by means of Automation from external programs, e.g. programs written in Visual Basic or C++ supporting Automation.

Such code can be used to control QlikView from other applications or from stand-alone programs.

Stand-alone executable files can be invoked from a QlikView document by means of launch buttons.

Internal Macro Interpreter

QlikView objects are also accessible via Automation from inside QlikView by means of the built-in macro interpreter.

Macros written in VBScript or JScript inside a QlikView document can currently be invoked in several ways:

Document events:

- 1 A macro can be run after opening a QlikView document.
- 2 A macro can be run after script re-execution.
- 3 A macro can be run after the **Reduce Data** command.
- 4 A macro can be run after a selection in any field in the document.

Sheet events:

- 5 A macro can be run after a sheet is activated.
- 6 A macro can be run when a sheet is deactivated.

Sheet object events:

- 7 A macro can be run after a sheet object is activated.
- 8 A macro can be run when a sheet object is deactivated.

Button events:

- 9 A button sheet object can be linked to a macro.

Field events:

- 10 A macro can be run after a selection has been made in a specified field.
- 11 A macro can be run when a selection is made in any field logically associated with a specified field.
- 12 A macro can be run when selections are locked in a specified field.
- 13 A macro can be run when selections are unlocked in a specified field.

Variable events:

- 14 A macro can be run after a value has been entered in a specified variable.
- 15 A macro can be run when the value of a specified variable containing a formula has been changed due to a change in the formula value.

26 INTERNAL MACRO INTERPRETER AND MACROS

26.1 Edit Module Dialog

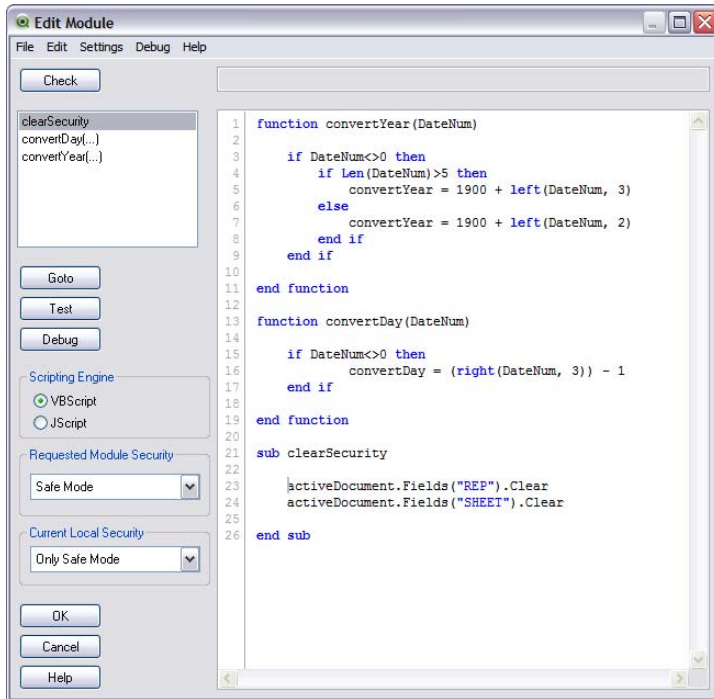


Figure 96. The Edit Module dialog

Macros and custom defined functions can be written in VBScript or JScript using the **Edit Module** dialog. The module is saved with the document.

By choosing **Edit Module** from the **File** menu, or by choosing the **Edit Module** button from the toolbar, you will enter the **Edit Module** dialog. The following controls are available:



The centerpiece of the dialog is a large edit box where you type your macros. All macros should be written as subroutines between a matching pair of `sub..end sub` (VBScript only) or as functions between a matching pair of `function..end function` (VBScript) or as `function..{..}` (JScript).

All subroutines and functions which have been syntax checked and found valid will be listed to the left in the dialog.

Check

Once you have written a macro, you can have it syntax-checked by pressing this button. All approved subroutines and functions will be listed to the left in the dialog. Functions and subroutines with parameters will be listed followed by (...).

Message

The current status and any error messages will be displayed in this box.

Goto

All subroutines and functions which have been syntax checked and found valid will be listed to the left in the dialog. By selecting a name in the list and pressing this button the macro edit box will be scrolled to the selected entry point.

Test

After selecting a macro in the list of entry points you can test its operation by clicking this button, provided that the macro is a VBScript subroutine without parameters. Error messages will be displayed in the **Message** box.

Debug

By pressing this button you will enter macro debug mode. Before pressing the button you must mark the macro that you wish to debug in the list of entry points. The macro selected must be a VBScript subroutine without parameters. You must also set one or more breakpoints in the code. Breakpoints are to be toggled by clicking on a line number next to a position in the code or by positioning the cursor in the line and pressing F9. See description of macro debugger below for further details.

Scripting Engine

Sets the scripting engine for the document. You may choose between **VBScript** and **JScript**.

Requested Module Security



Figure 97. The Module Security dialog

The designer of the QlikView document can set the intended macro security level to **Safe Mode** or **System Access**. By indicating **Safe Mode** the document designer indicates that the macros in the module do not contain any code that can access the system or applications outside QlikView. Typically this would mean code containing `CreateObject`, `GetObject` or `Launch`. If such code is encountered during macro execution in a document declared to be in **Safe Mode**, the execution will fail. If however the document designer indicates **System Access** mode the end user will be prompted (see Figure 97 above) when opening the document to approve system access (**Allow System Access**), disable all macros in the document (**Disable Macros**) or allow only macros without system access (**Safe Mode**). As soon as the user has chosen to approve or ban the macros this will be remembered by the system and no more prompts will appear when opening the document.

User Allowed Security Level

When opening a document declared to be in safe mode by the document's designer, which contains potentially unsafe code in the macro module script, the user will be prompted to approve, disable or partially disable macros (see above). This choice will be remembered by the system but can be changed at any later time via this setting. The macro security level can be set to **Don't run at All**, **Safe Mode** or **System Access**.

OK

This button saves changes and takes you back to the QlikView main menu.

Cancel

This button discards changes and takes you back to the QlikView main menu.

Furthermore, the **Edit Module** dialog contains a number of commands in five menus:

Export menu**Export to Module File...**

Saves the contents of the edit box in a text file to be specified in the **Save Module** dialog that appears. The file will have the extension .qvm.

Print...

Opens Windows standard **Print** dialog. This command can also be invoked via the shortcut command CTRL+P.

Edit menu**Undo**

Undoes the latest change. This command can also be invoked via the shortcut command CTRL+Z.

Redo

Redoes the latest **Undo**. This command can also be invoked via the shortcut command CTRL+Y.

Cut

Exports the selected text to the Clipboard. This command can also be invoked via the shortcut command CTRL+X.

Copy

Copies the selected text to the Clipboard. This command can also be invoked via the shortcut command CTRL+C .

Paste

Pastes the contents of the Clipboard into the dialog at the position of the cursor. This command can also be invoked via the shortcut command CTRL+V.

Clear

Clears all the text in the edit box.

Select All

Selects all the text in the edit box.

Insert File...

Lets you browse for a file containing macros and inserts its contents into the edit box at the position of the cursor.

Find/Replace

Opens a dialog allowing you to find and replace numbers or characters in the macros. This command can also be invoked via the keyboard shortcut command CTRL+F.

Goto...

Opens a dialog where you can enter the number of the page where you want to go. This command can also be invoked via the shortcut command CTRL+G.

Comment

Transform a text row, or text rows, in the module script into a comment.

Uncomment

Transforms the text row(s) back into original script text.

Settings menu

Configure

Opens the **Editor** page in the **User Preferences** dialog (page 99), where you can set the font and color of the different text types appearing in the edit box.

Debug menu

These commands are used to debug the script. In order to use the debugger, the Microsoft Script Debugger must be installed. See “Macro debugger” on page 320 for more information on the menu commands.

Help menu

Help

Opens the interactive QlikView help.

26.2 Macro debugger

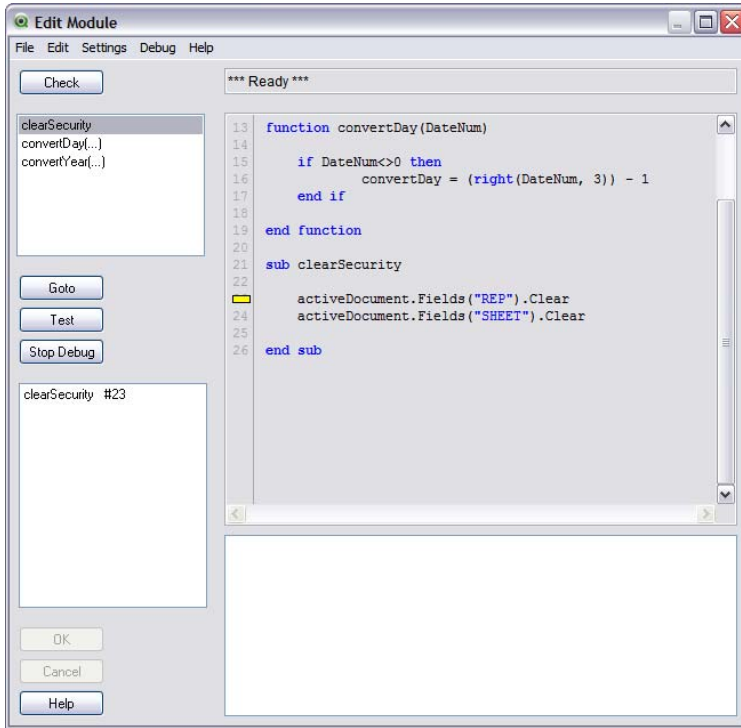


Figure 98. The Macro Debug page of the Edit Module dialog

Before you can use the macro debugger, you must install the Microsoft Script Debugger.

By pressing the **Debug** button in the **Edit Module** dialog you will enter macro debug mode. In this mode you may step through the macro and inspect variable contents as you go along.

Before debug mode is entered you must select a macro (subroutines without parameters only) from the list and set one or more breakpoints in the code. Breakpoints can be toggled at any time by clicking on a line number next to a position in the code or by positioning the cursor in the line and pressing F9. Pressing CTRL+SHIFT+F9 will clear all breakpoints.

When entering debug mode two new panes appear in the dialog window. In the lower left you will find a pane indicating the current call stack. By double-clicking in the list the editor window will be scrolled to the position of the call clicked upon. The execution pointer will not be moved however.

In the lower right you will find a pane with current variables. By double-clicking one of the variables you will open the **Variables** dialog where the properties of the variable can be further examined and its value changed interactively.

In debug mode a separate menu becomes available. Many of the items on this menu have a keyboard shortcut counterpart. The keyboard shortcuts can be individually customized in the **Editor** page of **User Preferences**. The shortcuts mentioned below are the installation defaults.

Continue

Continues execution of the macro to the next breakpoint or the end of the macro. Equivalent to pressing F5.

Step Into

Steps execution one line. If the next line is a call to a subroutine execution will be stepped to the first line of the subroutine. Equivalent to pressing F8.

Step Out

Executes the remaining statements of the current subroutine. Equivalent to CTRL+SHIFT+F8.

Step Next

Steps execution one line. If the next line is a call to a subroutine the subroutine will be executed in its entirety and stopped at the first line after the subroutine call. Equivalent to pressing SHIFT+F8.

Variable Window

Opens a window where the values of different variables can be inspected and changed.

Expression Window

Opens a dialog where you may enter and evaluate a VBScript or JScript expression based on current variable values.

Clear Breakpoints

Clears the breakpoints set in the code for debugging. Equivalent to pressing CTRL+SHIFT+F9.

Stop Debugging

Exits debug mode and returns to the normal **Edit Module** dialog.

26.3 Invoking Macros

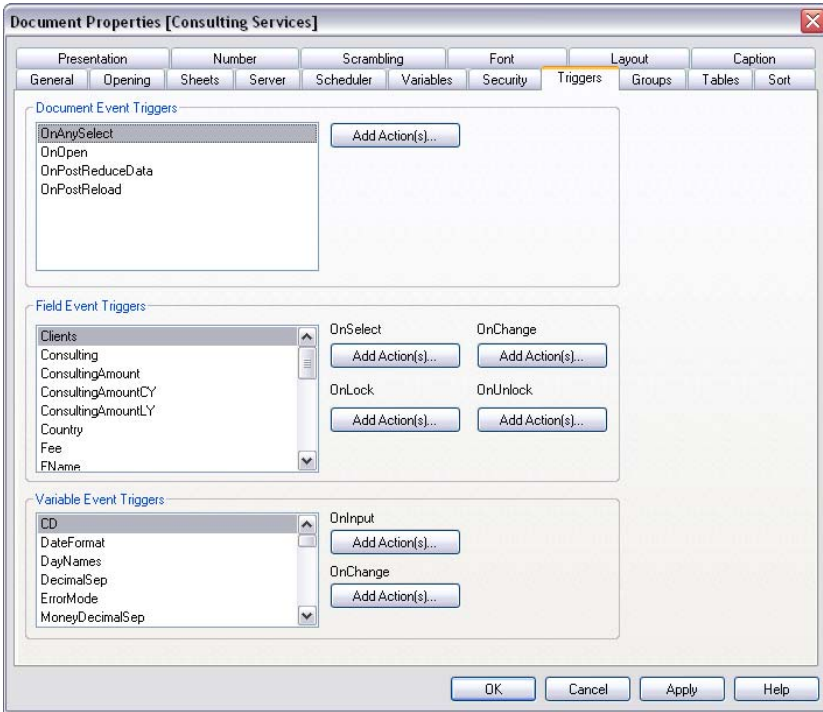


Figure 99. The Triggers page in the Document Properties dialog

Note There are limitations as to the behavior of certain macro triggers when working with documents on QlikView Server. Please study the special section "Considerations when using macros in QV documents on the QlikView Server" in the QlikView Server reference manual before designing server documents with macro triggers.

Invoking Macros on Document Events and Field Events

Choose the **Triggers** page in the **Document Properties** dialog.

Document Event Triggers

In the **Document Event Triggers** group you can set actions to trigger on selected events in the document. You must select an event in the list before you can assign actions or macros to it.

OnAnySelect

Select this event to assign an action to be executed each time a selection has been made in any field of the QlikView document.

OnOpen

Select this event to assign an action to be executed each time the QlikView document is opened.

OnPostReduceData

Select this event to assign an action to be executed after each time the **Reduce Data** command has been executed.

OnPostReload

Select this event to assign an action to be executed each time the script has been re-executed.

Add Action(s)

This button opens the **Actions** page. On this page you can add an action to the trigger. Choose the action **macro** to assign an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. See page 200 for how to add actions. The action (if it exists) will be executed each time the selected document event occurs. When an action has been assigned to the event, the button changes to **Edit Action** and you can change the action for the event.

Field Event Triggers

In the **Field Event Triggers** group you can set actions to trigger on changes in the logical state of a specified field in the document. Select a field in the list and press one of the buttons to assign an action to it. The buttons open the **Actions** page. Here you can assign an action, including an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. See page 200 for how to add actions. When an action has been assigned to the field, the button changes to **Edit Action**. The following **Field Event Triggers** exist:

OnSelect

The action will be executed each time a selection has been made in the specified field.

OnLock

The action will be executed each time the field is locked.

OnChange

The action will be executed each time a selection has been made in any field which is logically associated with the specified field.

OnUnlock

The action will be executed each time the field is unlocked.

Variable Event Triggers

In the **Variable Event Triggers** group you can set actions to trigger on changes in the contents of a specified variable in the document. Select a variable in the list and press one of the buttons to assign an action to it. The buttons open the actions page. On this page you can add an action to the trigger. Choose the action **macro** to assign an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. See page 200 for how to add actions. When an action has been assigned to the variable, the button changes to **Edit Action** and you can change the action for the event. The following **Variable Event Triggers** exist:

OnInput

The action will be executed each time a new value is directly entered in the selected variable.

OnChange

The action will be executed each time the value of the selected variable changes as a result of changes in other variables or the logical state of the document. This typically applies when the variable contains a formula.

Note For backward compatibility to earlier versions of QlikView, the action must consist of a macro action. An action that consists of only one macro action is translated back to old string format on save.

Invoking Macros on Sheet Events

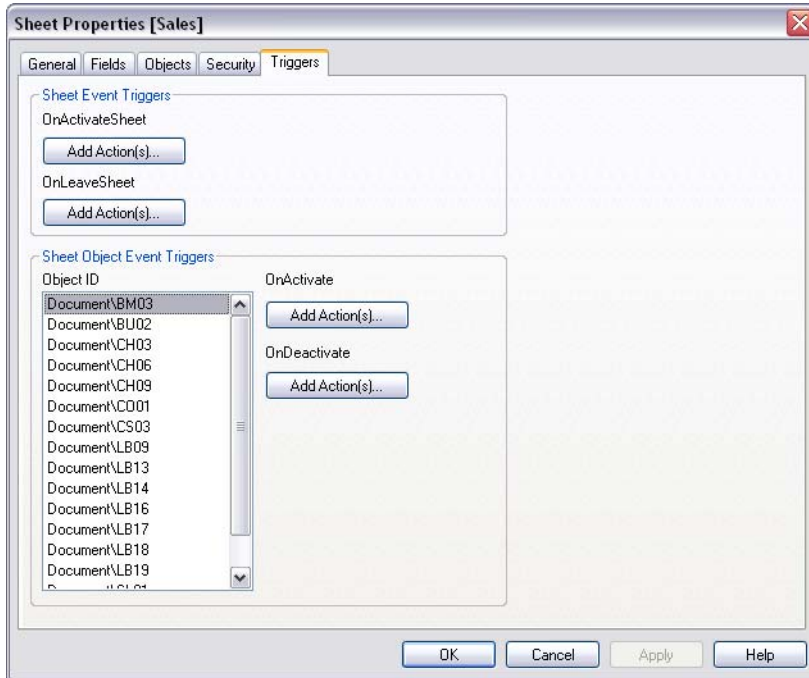


Figure 100. The Triggers page of the Sheet Properties dialog

Choose the **Triggers** page in the **Sheet Properties** dialog.

Sheet Event Triggers

In the **Sheet Event Triggers** group you can set actions to trigger on change of active sheet. Pressing the **Add Action(s)** button for either event brings you to the **Actions** page (see page 200 for how to add actions). On this page you can add an action to the event. Choose the action **macro** to assign an existing macro name or type any name for which you later can create a macro in the **Edit Module** dialog. When an action has been assigned to the event, the button changes to **Edit Action** and you can change the action for the event. The following **Sheet Event Triggers** are configurable:

OnActivateSheet

The action will be executed each time the sheet is activated.

OnLeaveSheet

The action will be executed each time the sheet is deactivated.

Sheet Object Event Triggers

In the **Sheet Object Event Triggers** group you can set actions to trigger on activation and deactivation of a specified sheet object on the sheet. Select an **Object ID** and choose the event to act as a trigger. Click on the **Add Actions** button to add an action . See page 200 for how to add actions.

OnActivate

The action will be executed each time the sheet object is activated.

OnDeactivate

The action will be executed each time the sheet object is deactivated.

Invoking Macros With Sheet Object Buttons

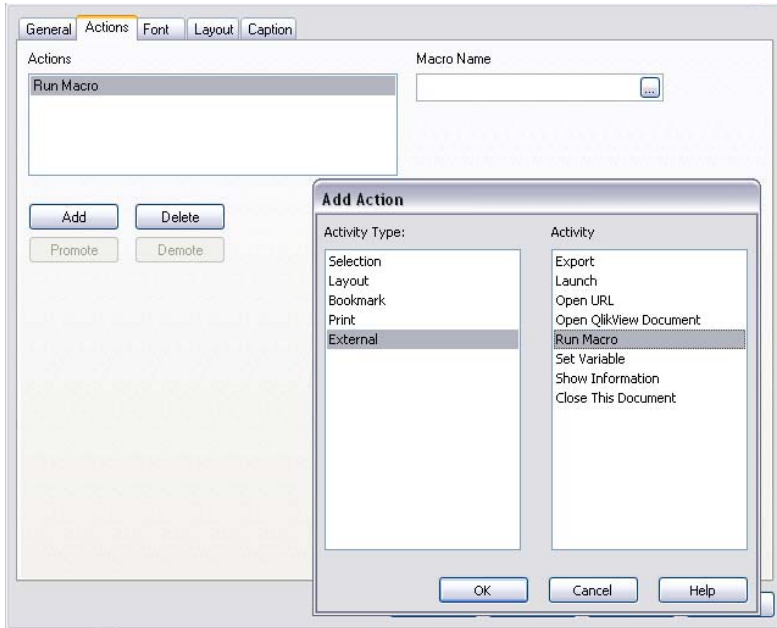


Figure 101. Choosing the Run Macro action

Choose the **Actions** page of the **Button Properties** dialog and **Add**. In the **Add Actions** dialog that opens, choose the action **Run Macro** from the **Action Type External**.

Macro Name

Enter the path and name of the macro to be run. You can type any name for which you later create a macro in the **Edit Module** dialog. The macro (if it exists) will be executed when the button is clicked.

26.4 Examples of Macros

This section gives you an example of a VBScript macro, which can be very useful.

Using VBScript Input Boxes for User Interaction

```
rem ** Clear selections, ask for product, **
rem ** select that product, **
rem ** go to sheet "Market", **
rem ** show pivot table "Sales" **
Sub ChooseValue
    Set q = ActiveDocument
    q.ClearAll (false)
    Set f = q.Fields("Model Name")
    x = inputbox ("Enter product")
    f.Select(x)
    set s = q.Sheets("Market")
    s.Activate
    s.SheetObjects("Sales").Activate
end sub
```

26.5 Special library functions for JScript

Whereas the standard VBScript functions `InputBox` and `MsgBox` can be used freely in VBScript macros, no direct counterpart is available when using JScript. For this purpose a special library `qvlib` has been added for these actions. The functions are demonstrated in the example below.

```
// JScript
function Test()
{
    personName = qvlib.InputBox("What is your name?")
    msg = "Hello " + personName + " !"
    qvlib.MsgBox(msg)
}
```

The `qvlib` functions actually work also in VBScript macros as shown in the example below, but in this case you might as well use the generic VBScript functions `InputBox` and `MsgBox`.

```
rem VBScript
sub TestModule
    name = qvlib.InputBox("What is your name?")
    msg = "Hello "
    msg = msg + name + " !"
    qvlib.MsgBox(msg)
end sub
```

27 GETTING HOLD OF A QLIKVIEW DOCUMENT

27.1 Accessing QlikView Documents from the Outside

QlikView documents can be accessed from outside the QlikView program by means of the Automation interface.

Accessing an Existing QlikView Document

A direct way to access existing documents is to use `GetObject`.

Example (Does not work in VBScript!):

```
Private Sub OpenAndReload_Click()
    Set QvDoc = GetObject
        ("c:\windows\desktop\test.qvw")
    QvDoc.Reload
End Sub
```

27.2 Accessing QV Documents from the Internal Macro Interpreter

When using the internal macro interpreter, the only reference available is the `ActiveDocument` property of the class `Application`. All references must be made starting from that point.

Examples:

```
Sub C1r
    Set QVDoc = ActiveDocument
    QVDoc.ClearAll(false)
End sub
Sub EndQV
    ActiveDocument.Application.Quit()
End sub
```



28 VBSCRIPT FUNCTION CALLS FROM SCRIPT

VBScript or JScript functions defined in the macro module of a QlikView document can be called from the script. If a function called is not recognized as a standard script function, a check will be made to see if a custom functions resides in the module. This gives you a large degree of freedom to define your own functions. Using custom macro functions will of course be somewhat slower than executing the standard functions.

```

rem *****
rem ***THIS IS VBSCRIPT CODE FROM THE MODULE*****
rem*****
rem ***** Global variables *****
dim flag

rem *****functions accessible from script *****

rem ***** wrap for input box *****
function VBin(prompt)
    VBin=inputbox(prompt)
end function

rem ***** clear global flag *****
function VBClearFlag()
    flag=0
end function

rem ***** test if reference has passed *****
function VBRelPos(Ref, Current)
    if Ref=Current then
        VBRelPos="Reference"
        flag=1
    elseif flag=0 then
        VBRelPos="Before "&Ref&" in table"
    else
        VBRelPos="After "&Ref&" in table"
    end if
end function

// *****
// *****THIS IS THE SCRIPT *****
// *****

let MaxPop=VBin('Max population in millions :');
// Ask limit

let RefCountry=VBin('Reference country :');
// Ask ref.

let dummy=VBClearFlag(); // Clears the global flag

Load
    Country,recno(),
    Capital,
    "Area (km.sq) ",

```



```
"Population(mio) ",  
VBrelPos ('$(RefCountry) ', Country)  
as RelativePos  
from country1.csv  
(ansi, txt, delimiter is ',', embedded labels)  
where "Population(mio)" <= $(MaxPop);
```

28.1 Transfer of Parameters

The following rules apply for parameter transfer of parameters between the load script and VBScript:

- Missing parameters are passed as NULL
- If the actual expression evaluates to a valid number, the number is passed, else if the actual expression evaluates to a valid string, the string is passed, else NULL is passed.
- NULL is passed as VT_EMPTY.
- Return values are treated in a natural way.



29 THE VISUAL BASIC PROGRAMMING LANGUAGE

29.1 About Visual Basic

This documentation is not intended as a manual for the Visual Basic language. The reader is supposed to possess a basic knowledge of Visual Basic programming.

Visual Basic is a very suitable language for writing macros and other programs for controlling QlikView via its Automation interface. Such code could of course also be written in other programming languages supporting Automation, e.g. Visual C++, but for most users Visual Basic in one of its forms is likely to be the first choice. For this reason a few words about the language could be appropriate in this documentation.

In comparison with its ancestor, BASIC, Visual Basic however has a number of additions which make it a quite modern object oriented programming language. Visual Basic has during the last few years turned into a macro and scripting language widely used by major software producers. It was therefore a natural choice to use this language (in its VBScript incarnation) as the macro language for QlikView.

29.2 Differences Between VB, VBA and VBS

Visual Basic as a stand alone compiler (VB), Visual Basic for Applications (VBA) and Visual Basic Script (VBS) are all variations of the Visual Basic language. Most of the basic syntax and keywords are the same, but small variations exist. When it comes to advanced capabilities and the ability to communicate with the outside world from the code, there are large differences, especially between VB/VBA on one side and VBS on the other. Some main differences between VB, VBA and VBScript are described below.

	VB	VBA	VBScript
Compile stand-alone.exe	yes	no	no
Full VB syntax	yes	yes	no
Data types	full	full	small

Different Versions of VBScript

Apart from the incomplete listing of differences above, additional differences may exist as a result of which version of VBScript you are using. QlikView is currently shipped with VBScript version 3.1.

VB Features Not Included in VBScript

Clipboard access
Collection
Conditional compilation
Constants
Data types (only variant supported)
Date/Time
DDE
Debug commands
File input/output
Financial

Features with Limited Implementation in VBScript

Array handling
Error handling (only On Error Resume Next)

Some VB Keywords and Functions Not Supported by VBScript

Declare
Dim ... as <datatype>
DoEvents
For Each ... Next
GoSub ... Return
GoTo
Line Numbers
Line Labels
Like
New
With ... End With

29.3 Note About Parameters to Methods and Properties

In VB, VBA and VBS the following rules apply to the use of parentheses around arguments:

- Always use parentheses around parameters to properties.

- Arguments to single parameter methods may optionally be enclosed in parentheses.
- Arguments to multiple parameter methods *must not* be enclosed in parentheses.

29.4 Getting Hold of the System from VBScript

When using the internal VBScript interpreter, it is possible to get hold of a reference to the local file system by using a statement like the following:

```
set fso = CreateObject("Scripting.FileSystemObject")
```

This reference can then be used for operations towards the file system.

Example:

```
sub LogFunktion
' This routine logs selection to a text file

set fso= CreateObject("Scripting.FileSystemObject")
set mypath = ActiveDocument.GetProperties
directory = mypath.MyWorkingDirectory

On Error Resume Next

' See if file already exists.
set f = fso.GetFile(directory & "log.txt")
' If not, then create it.
if Err <> 0 then
set f=fso.CreateTextFile(directory & "log.txt")
end if

set txsStream = f.OpenAsTextStream(8) 'For Append

set doc = ActiveDocument
set mySel = doc.fields("Field").GetSelectedValues

for i = 0 to mySel.Count - 1
txsStream.WriteLine Now& " "&mySel.Item(i).text
next
txsStream.WriteBlankLines 1
txsStream.Close

end sub
```



30 USING MACROS IN QV DOCUMENTS ON THE QV-SERVER

30.1 Macros on QlikView Server

QlikView Server can execute macros in QlikView documents. Due to client-server environment some considerations must be taken when using macros.

When using Java clients all macros are run in the server. This causes considerable restrictions in permitted operations.

When using QlikView as a client all macros are executed in the client. In this case more operations can be permitted.

30.2 Macro functionality limitations

Functionality that will normally work well in macros in the QlikView Server environment with any type of client is:

- Logical operations such as clearing or selecting in fields
- Operations related to variables

The following types of functionality will work when using QlikView as client but NOT with Java clients:

- Layout operations acting on sizing, positioning, minimization etc. of sheet objects using direct access functions rather than SetProperties. Examples: SetRect, Minimize, Restore
- Activate and Deactivate on sheets and sheet objects
- Operations changing User Preferences

The following types of functionality are not to be used in the QlikView Server environment, as they may cause unexpected results:

- Layout operations acting on the properties of sheets and sheet objects via SetProperties
- Operations changing document or user settings
- All operations related to the script, including Reload
- Data reduction operations, e.g. ReduceData

-
- Operations such as Save and Open document

30.3 Macro trigger limitations

The following triggers will work as usual in the QlikView Server environment regardless of which client is used:

- Document.OnAnySelect
- Field.OnSelect
- Field.OnChange
- Field.OnUnlock
- Document.OnPressMacroButton
- Variable.OnChange
- Variable.OnInput

The following triggers will be meaningless in conjunction with the QlikView Java Objects client, but work with all other clients:

- Sheet.OnActivateSheet
- Sheet.OnLeaveSheet

The following types of functionality will work when using QlikView as client but NOT with Java clients:

- SheetObject.OnActivate
- SheetObject.OnDeActivate

The following triggers are not to be used in the QlikView Server environment, since they lack meaning or may cause unexpected results:

- OnOpen
- OnPostReduceData
- OnPostReload

30.4 VBScript functions

VBScript functions defined in the module of a QlikView document will generally work well on QlikView Server. General functionality limitations as defined in the previous section apply.

30.5 Server Side Export

It is possible to export the content of any chart to a text file on the server by using a Macro connected to a Button. This feature has to be added in the QlikView document

and then the macro can be executed from the QlikView Server directly in the web browser.

In the QlikView document QWT.qvw provided on the CD, you find a sheet called "Server Side Export". As the button is clicked, the content of the Straight Table CH20 can be exported as a text file to the server directory for QlikView Server. The exported data can then easily be read into Excel or other programs.

Below you find a sample of VB Script code used in the Edit Module in QlikView for making the export possible. The macro is then connected to a trigger, in the case of this example, a button.

```
Sub ServerExportEx
    Set st = ActiveDocument.GetSheetObject("CH20")
    st.ServerSideExportEx "C:\ProgramFiles\QlikView
Server\CH20export.skv" , ";" , 0
End Sub
```



INDEX

A

- Activating
 - sheet73
 - sheet objects92
- Average (statistics box)131

B

- Bookmark object237–242
- Border61
 - of sheet objects114
 - settings in document properties61
- Button195–209
- Button actions200

C

- Calendar Mode225
- Caption
 - in sheet objects68, 120
 - settings in document properties68
- Chart135
- Color Area dialog23–25
- Compression16
- Constraints
 - in input box189
- Copy74
- Copying
 - field values to clipboard272
 - image to clipboard74
- Creating
 - a bookmark object237
 - a button195
 - a current selections box173
 - a line/arrow object219
 - a list box97
 - a multi box137
 - a search object243
 - a sheet73
 - a sheet object91
 - a slider object226
 - a statistics box125
 - a table box155
 - a text object211
 - an input box181

- Current Selections box173–179
- Custom Format Cell Dialog160
- Custom object251–255

D

- Data
 - types291
- Date format289
- Default Export Encoding20
- Disable Layout Undo19
- Document Event Triggers46, 322
- Document Properties15–71

E

- Edit Module dialog314
- Example63
- Exporting
 - image to file75
 - with button207

F

- Field Event Triggers46, 323
- Field groups48
- Fonts60
- Format code308
- Format codes300
- Fractile (statistics box)132

G

- Groups of fields48

H

- Help Text
 - in sheet objects123
 - setting ~ in document properties71

I

- Image border definition file63
- Input39
- Input box181–193
 - using181

Internal313
invoking322

K

Kurtosis (statistics box) 132

L

Layer61
Legacy Fractile Calculation 19
Line/arrow object 219–223
List box97–124
Local Objects and Server Objects91
Loosely Coupled Tables51

M

Macro debugger320
Macros322
 examples327
Macros, invoking
 on sheet events87, 325
 with button327
Median (statistics box)132
Memory
 statistics17
Missing count (statistics box)131
Multi box137–153

N

Null count (statistics box)131
Number format
 codes300
 dialogs300
 handling ~ in QlikView289–310
Number Interpretation291
Numeric count (statistics box)131

O

Object74
Object menu
 for button195
 for current selections box173
 for input box182
 for line/arrow object219
 for list box97
 for multi box137
 for sheet objects92
 for sheets74
 for slider object226

 for statistics box125
 for table box156
 for text object211
Only value (statistics box)132

P

Pop-up Window Settings21
Predefined Values41
Print Options263–271
Print Preview271
Print Sheet271

R

Rounded63

S

Search Object243
Search object243–249
Security page
 Document Properties42
 Sheet Properties85
Sheet Object Style20
Sheet Properties75
Skewness (statistics box)132
Slider Mode225
Slider/Calendar object225–236
Special328
Standard deviation (statistics box)131
Statistical functions131
Statistics box125–134
Styling Mode20
Sum (statistics box)131

T

Table box155–172
 using155
Tabrow20
Tabrow Style20
Tabs73
Text count (statistics box)131
Text object211–218
Theme Maker Wizard278
Time format289
Total count (statistics box)131
Triggers, invoking
 on document events45

U
Use61

V
Value40
Variable Event Triggers47, 324

Z
Zoom77



Reference Manual

Book III: Charts

Version 9.0 for Microsoft Windows®

Third Edition, Lund, Sweden, February 2010

Authored by QlikTech International AB /HIC/KHN/JNN/MSJ/CEN

Copyright © 1994-2010 Qlik®Tech International AB, Sweden.

Under international copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written permission of QlikTech International AB, except in the manner described in the software agreement.

Qlik®Tech and Qlik®View are registered trademarks of QlikTech International AB.

Microsoft, MS-DOS, Windows, Windows NT, Windows 2000, Windows 2003, Windows XP, Windows Vista, SQL Server, FoxPro, Excel, Access and MS Query are trademarks of Microsoft Corporation.

IBM, AS/400 and PowerPC are trademarks of International Business Machines Corporation.

Borland, Paradox and dBASE are trademarks of Borland International.

ORACLE and SQL*Net are trademarks of Oracle Corporation.

MacOS is a trademark of Apple Corporation.

Sun Java is a trademark of Sun Microsystems, Inc.





OVERVIEW BOOK III

CHARTS

GLOSSARY

APPENDIX

—

GLOSSARY

APPENDIX



CONTENT

PART I: CHARTS

1	INTRODUCTION	15
	1.1 Creation	18
	1.2 Chart Object Menu	18
	1.3 Selections in Charts	19
	1.4 Fast Chart Type Change	19
	1.5 Sizing and Moving Chart Components	19
	1.6 Chart Properties	20
2	BAR CHART	27
	2.1 Creation	27
	2.2 Bar Chart Object Menu	27
	2.3 Chart Properties	30
3	LINE CHART	87
	3.1 Creation	87
	3.2 Line Chart Object Menu	87
	3.3 Chart Properties	90
4	COMBO CHART	99
	4.1 Creation	99
	4.2 Combo Chart Object Menu	99
	4.3 Chart Properties	102
5	RADAR CHART	113
	5.1 Creation	113
	5.2 Radar Chart Object Menu	113
	5.3 Chart Properties	116
6	SCATTER CHART	127
	6.1 Creation	127
	6.2 Scatter Chart Object Menu	127
	6.3 Chart Properties	129
7	GRID CHART	145
	7.1 Creation	145
	7.2 Grid Chart Object Menu	145
	7.3 Chart Properties	147

8	PIE CHART	157
	8.1 Creation	157
	8.2 Pie Chart Object Menu	157
	8.3 Chart Properties	160
9	FUNNEL CHART	167
	9.1 Creation	167
	9.2 Funnel Chart Object Menu	167
	9.3 Chart Properties	170
10	BLOCK CHART	177
	10.1 Creation	177
	10.2 Block Chart Object Menu	177
	10.3 Chart Properties	180
11	GAUGE CHART	189
	11.1 Creation	189
	11.2 Gauge Chart Object Menu	189
	11.3 Chart Properties	191
12	PIVOT TABLE	201
	12.1 Creation	201
	12.2 Using the Pivot Table	201
	12.3 Pivot Table Object Menu	205
	12.4 Chart Properties	209
13	STRAIGHT TABLE	227
	13.1 Creation	227
	13.2 Using the Straight Table	227
	13.3 Straight Table Object Menu	228
	13.4 Chart Properties	231
14	EDIT EXPRESSION	249
	14.1 The Colormix Wizard	254
15	THE BOX PLOT WIZARD	261
	15.1 Starting the Box Plot Wizard	261
	15.2 Pages of the Box Plot Wizard	262
16	THE QUICK CHART WIZARD	265
	16.1 Starting the Quick Chart Wizard	266
	16.2 Pages of the Quick Chart Wizard	267
17	THE TIME CHART WIZARD	277
	17.1 Starting the Time Chart Wizard	278
	17.2 Pages of the Time Chart Wizard	279

18 THE STATISTICS CHART WIZARD	287
18.1 Starting the Statistics Chart Wizard	287
18.2 Chi2-Test	288
18.3 Paired t-test	289
18.4 Independent Samples t-test	291
19 CHART EXPRESSIONS	293
19.1 Chart Expression Syntax	293
19.2 Variable Text Expansion	294
19.3 Operators	295
19.4 Aggregation Functions	298
19.5 Synthetic Dimension Functions	361
19.6 Other Functions	362
20 EXAMPLES	449
20.1 Examples of Aggregate Qualifiers	449
20.2 Examples of Ranking Functions	450
20.3 Examples of Chart Inter-Record Functions	453
21 NESTED AGGREGATIONS AND RELATED ISSUES	459
21.1 Nested Aggregations with total Qualifier	459
21.2 Nested Aggregations with the aggr Function	459
21.3 Sum of rows in Pivot Tables	461
21.4 Linear Regression in Table Charts	464
22 CALCULATED FORMULAS	467
22.1 Entering a Calculated Formula	467
22.2 Expression Syntax for Calculated Formulas	467
22.3 Error Messages	468
23 FIELD GROUPS	469
23.1 Hierarchic Groups (Drill-Down)	469
23.2 Non-Hierarchic Groups (Cyclic)	470
24 CUSTOM ERROR MESSAGES	471

GLOSSARY

GLOSSARY	475
-----------------------	------------

APPENDIX

LIMITATIONS AND REQUIREMENTS	487
A.1 Limitations of the Amount of Data to be Loaded	487
A.2 System Requirements	487
A.3 Dynamic Link Libraries (dll's) Needed	489
KEYBOARD COMMAND SHORTCUTS	491
B.1 File Menu Command Shortcuts	491
B.2 Edit Menu Command Shortcuts	492
B.3 View Menu Command Shortcuts	492
B.4 Selections Menu Commands Shortcuts	493
B.5 Settings Menu Command Shortcuts	493
B.6 Bookmarks Menu Command Shortcuts	493
B.7 Tools Menu Command Shortcuts	494
B.8 Object Menu Command Shortcuts (List box, Statistics box and Open Multi Box)	494
B.9 Script Keyboard Shortcuts	494
B.10 F Key Keyboard Shortcuts	495
APPLICATION PERFORMANCE OPTIMIZATION	497
C.1 Introduction	497
C.2 Count (Distinct 'FieldName')	497
C.3 If (Condition(Text),....)	498
C.4 Sum (If (Condition, 'FieldName' ...))	499
C.5 If (Condition, Sum('FieldName')..)	500
C.6 If (Condition1, Sum('FieldName'), If (Condition2, Sum('Field- Name').....)	500
C.7 Sorting text	501
C.8 Dynamic captions and text objects	501
C.9 Macro triggers (“on change“)	502
FREQUENTLY ASKED QUESTIONS	503
D.1 Installation	503
D.2 QlikView Documents	504
D.3 Scripts and Loading Data	505
D.4 QlikView Logic	507
D.5 Layout	509
D.6 Sharing QlikView Documents with Other People	510
DATA PROTECTION ISSUES	513
E.1 Data Security and Integrity	513

E.2 The Right to Alter Data	513
E.3 The Right to View Data	514
E.4 Data Integrity	514

USING QLIKVIEW DOCUMENTS AS AN INFORMATION SERVER.... 515

F.1 Information Server	515
F.2 Extracting Data via ODBC/OLEDB	515
F.3 Extracting Data From AS/400 With Router	516
F.4 Extracting Data From Mainframe, AS/400 etc. Without Router	516
F.5 Using QlikView Binary Files	516
F.6 Aggregation	517
F.7 Sharing QlikView Documents in Workgroups	517
F.8 Mailing QlikView Documents	517
F.9 Posting QlikView Documents on a Web Page	517
F.10 Posting QlikView Documents on an FTP Server	518
F.11 QlikView Document on Server - Basic Setup	519
F.12 QlikView Document on Server - Advanced Setup	520
F.13 Customized QlikView Documents	521
F.14 QlikView Server	521
F.15 QlikView Publisher	522

BACKUS-NAUR FORMALISM..... 523





PART I: CHARTS

- **Creation of Charts**
- **Chart Properties**
- **Chart Expressions**
- **Calculated Formula**
- **Field Groups**



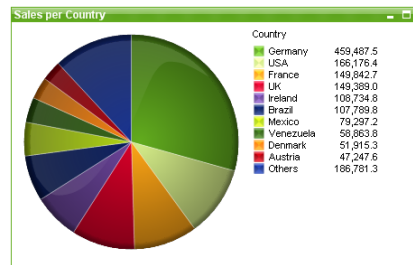
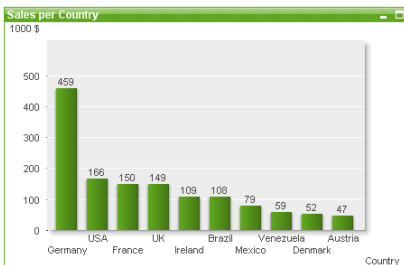
1 INTRODUCTION

Charts and tables are sheet objects that can show numbers very compactly. You can e.g. show sums of money, distributed over different fields such as year, month, account number, etc.

Charts and tables can be set to display either the frequencies of different values of a field, or a calculated entity, e.g. the sum of the possible values of a field. In both cases, a certain field must be chosen as x-axis, i.e. this field will be used to label the slices of the pie, the different bars in the bar chart and the rows in the pivot table, respectively.

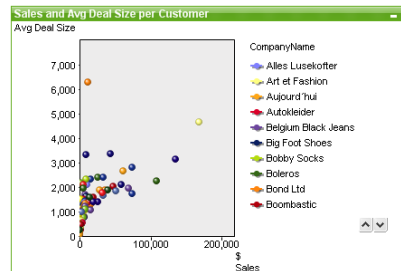
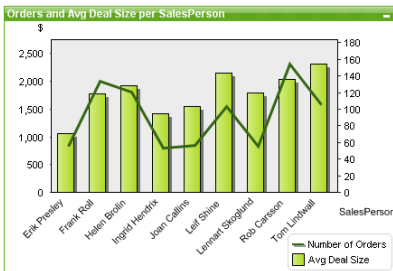
The different chart types available

The chart types you can choose between are bar chart, line chart, combo chart, scatter chart, pie chart, pivot table, straight table, radar chart, grid chart, block chart and gauge chart.:



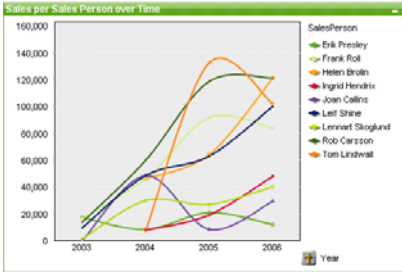
Bar chart

Pie chart

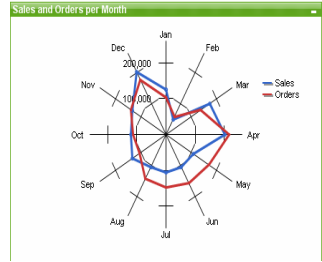


Combo chart

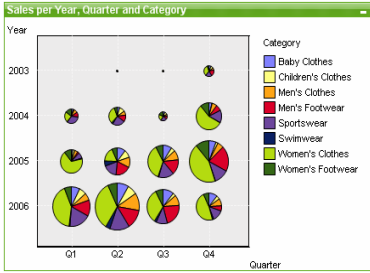
Scatter chart



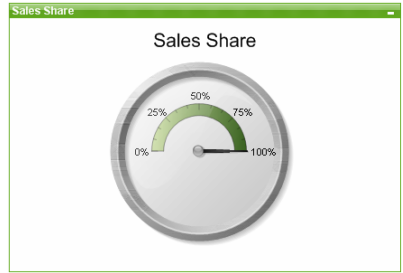
Line chart



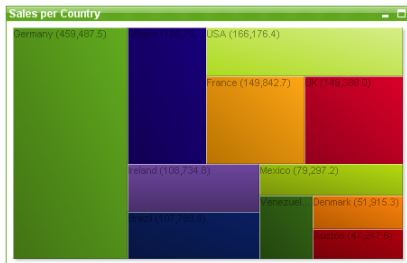
Radar chart



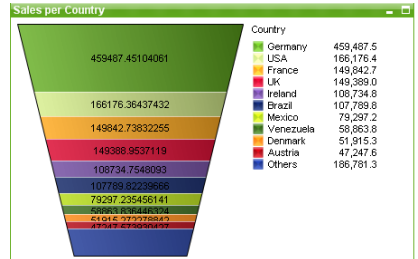
Grid chart



Gauge chart



Block chart



Funnel chart

Sales per CategoryName				
CategoryName	ProductName	Sales	Quantity	
Men's Clothes	Atlas Lussekkorta	\$30,126.55	1057	
	Bow tie	\$9,534.57	1315	
	Desperado Jeans	\$18,240.68	706	
	Lenin Jeansshorts	\$14,900.64	628	
	Mi2 Trousers	\$17,944.48	1067	
	O-Man Underwear	\$1,649.87	298	
	Rossi Bermuda Shorts	\$10,947.25	1367	
	Samba Soccer Socks	\$4,941.14	1175	
	US-Master Jeans	\$21,764.94	817	
	Total	\$130,050.12	8660	
Women's Clothes	Chantell Shirt	\$7,504.70	388	
	Halter Dress	\$361,096.85	981	
	Jack Flash Dress	\$42,638.00	722	
	Largoste Shirt	\$4,433.35	246	
	Le Baby Dress	\$47,571.88	623	
	Minnki Palsii	\$10,472.71	184	
	Okkaba Skin Jackets	\$42,258.78	601	
	Oyaki Kimono	\$9,084.42	806	

Sales per CategoryName			
CategoryName	ProductName	Sales	Quantity
Men's Clothes	Atlas Lussekkorta	\$1,665,925.34	51962
Men's Clothes	Bow tie	\$30,126.55	1057
Men's Clothes	Desperado Jeans	\$9,534.57	1315
Men's Clothes	Lenin Jeansshorts	\$18,240.68	706
Men's Clothes	Mi2 Trousers	\$14,900.64	628
Men's Clothes	O-Man Underwear	\$17,944.48	1067
Men's Clothes	Rossi Bermuda Shorts	\$1,649.87	298
Men's Clothes	Samba Soccer Socks	\$10,947.25	1367
Men's Clothes	US-Master Jeans	\$4,941.14	1175
Women's Clothes	Chantell Shirt	\$21,764.94	817
Women's Clothes	Halter Dress	\$7,504.70	388
Women's Clothes	Jack Flash Dress	\$361,096.85	981
Women's Clothes	Largoste Shirt	\$42,638.00	722
Women's Clothes	Le Baby Dress	\$4,433.35	246
Women's Clothes	Minnki Palsii	\$47,571.88	623
Women's Clothes	Okkaba Skin Jackets	\$10,472.71	184
Women's Clothes	Oyaki Kimono	\$42,258.78	601
Women's Clothes		\$9,084.42	806

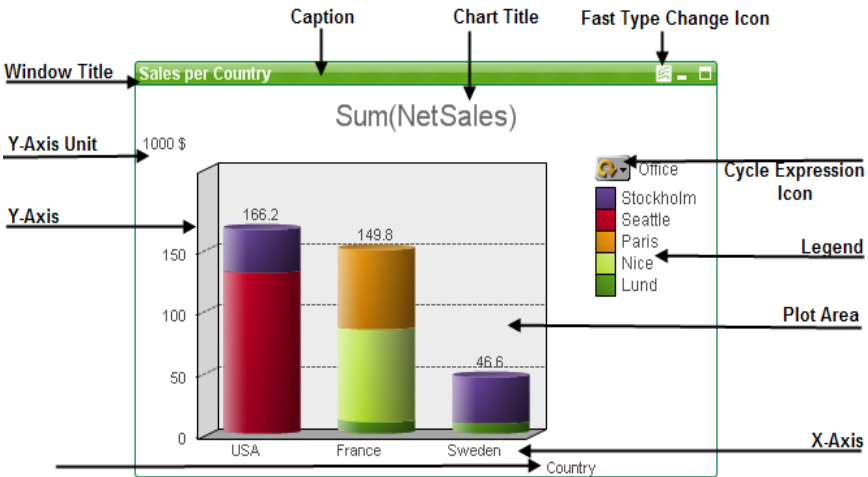
Pivot table

Straight table

QlikView charts can be divided into two major categories. The first, graph charts, consists of bar, line, combo, pie, scatter, radar, grid, block and gauge charts. The second category, table charts, consists of straight tables and pivot tables. These chart types are drawn as tables with cells in columns and rows. Please note that table boxes, although in many respects similar to table charts are not charts but a separate type of sheet objects.

Graph Charts

The bar chart below shows some of the more common components of a QlikView graph chart:



The position of the different components can in many cases be altered by the user. See chapter "Sizing and Moving Chart Components" on page 19 for details.

Table Charts:

Account Group	Account Description	Cash Flow Drill	Forecast (as is)	Budget	Variance	Forecast (Simulation)	Budget
6520 Postage		Cash Flow aus B...	46.552	35.370	-32%	46.552	35.370
6540 Promotion and entertain...		Cash Flow aus B...	197.817	77.522	-155%	197.817	77.522
6560 Rent, office		Cash Flow aus B...	149.109	375.707	60%	149.109	375.707
6590 Repairs and maintenance		Cash Flow aus B...	44.299	26.229	-89%	44.299	26.229
6600 Shipping supplies		Cash Flow aus B...	81.024	51.793	-56%	81.024	51.793
6620 Shop supplies		Cash Flow aus B...	51.986	24.110	-116%	51.986	24.110
6640 Subcontract costs		Cash Flow aus B...	22.280	16.895	-32%	22.280	16.895
6660 Traveling expenses		Cash Flow aus B...	84.115	54.996	-53%	84.115	54.996
6680 Telephone, telex, fax		Cash Flow aus B...	112.998	96.362	-17%	112.998	96.362
6700 Utilities		Cash Flow aus B...	57.442	37.086	-55%	57.442	37.086
6720 Wages & benefits, direct		Cash Flow aus B...	280.136	839.682	67%	280.136	839.682
6740 Wages & benefits, indirect		Cash Flow aus B...	51.659	123.326	58%	51.659	123.326
6760 Wages casual, direct		Cash Flow aus B...	40.848	115.299	65%	40.848	115.299
Total			2,051,337	3,031,724	32%	2,051,337	3,031,724
Cost of Sales			1,285,848	1,405,568	9%	1,285,848	1,405,568
Other Inco...			-2,001,455	522,894	483%	-2,001,455	522,894
Provision...			14,789	42,048	65%	14,789	42,048
Revenue			-3,374,727	-4,035,386	16%	-3,374,727	-4,035,386
Total			-2,024,208	966,648	309%	-2,024,208	966,648

The pivot table above shows some of the more common components of a QlikView table chart.

1.1 Creation

Charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens a dialog that will help you create the chart.



If you want to create a simple chart and find the great number of different settings available overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu, or click the **QuickChart** button in the toolbar. The wizard that appears will help you to create a chart in a quick and easy way. The result is however always a full-blown chart - it is only the creation process that has been simplified.



For a detailed description of the Quick Chart Wizard, see chapter “The Quick Chart Wizard” on page 265.

1.2 Chart Object Menu

The chart **Object** menu is found as the **Object** menu when a chart is active. It can also be opened as a context menu by clicking with the right mouse button on a chart. The

chart object menu looks different depending on the active chart type. To find the commands that apply to the chart type you are working with, see the chapter on that specific chart.

1.3 Selections in Charts

Selections in field data can be made directly in all of QlikView's charts and tables by means of clicking or painting with the mouse. For details about selections in charts, see chapter “Selections in Other Objects” on page 138 in Book I.

1.4 Fast Chart Type Change

The type of chart can be changed by the user without going via the **Chart Properties** dialog, provided that the **Fast Type Change** option has been selected on the **General** page of the **Chart Properties** dialog.

An icon showing the next available chart type (**Allowed Types** selected for fast type change) will appear in the chart. By left-clicking the icon, the chart will change to the indicated type. By right-clicking the icon you get a drop-down menu with all selected types.



The application designer can choose **Preferred Icon Position** of the fast type change icon. By choosing **In Caption** the icon will appear in the chart caption, provided that a caption is shown. By choosing **In Chart** the icon will appear inside the chart, provided that the chart is not a pivot table or straight table. If the preferred position is not available, QlikView will try to use the other option. In table charts without caption, no icon will be shown.

1.5 Sizing and Moving Chart Components

In QlikView's graphical charts many of the individual chart components can be sized or moved according to individual wishes. By pressing the SHIFT and CTRL keys simultaneously and keeping them depressed while a chart is active you will ENTER the chart layout edit mode. In this mode thin red rectangles will appear around those components of the chart, that can be sized or moved. You may then use the mouse drag and drop technique to move things around.

The following components can be changed:

The **Chart Title** can be both moved and sized. It can be docked to the top, bottom, left and right border of the chart and also be positioned free-floating anywhere you please within the chart.

The **Chart Legend** can be both moved and sized. It can be docked to the top, bottom, left and right border of the chart and also be positioned free-floating anywhere you please within the chart.

The area taken up by the **Chart Axes** and their labels can be sized.

A **Cycle Expression Icon** can be moved to a free-floating position anywhere you please within the chart.

Chart Texts can be moved to a any position in the chart and they can be sized.

A **Fast Chart Type Change Icon** can be moved to a free-floating position anywhere you please within the chart.

The **Plot Area** itself cannot be sized or moved but will be drawn in the space remaining between the axes and docked legend and title.

1.6 Chart Properties

When you click the **Create Chart** button in the toolbar, the chart properties dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The settings available in the **Chart Properties** dialog's different properties pages depend on the chart type you have chosen on the first page (**General** page). For information on the different pages of the **Chart Properties** dialog (except for the **General** page, which is described below), we therefore refer to the chapters on the specific charts.

General

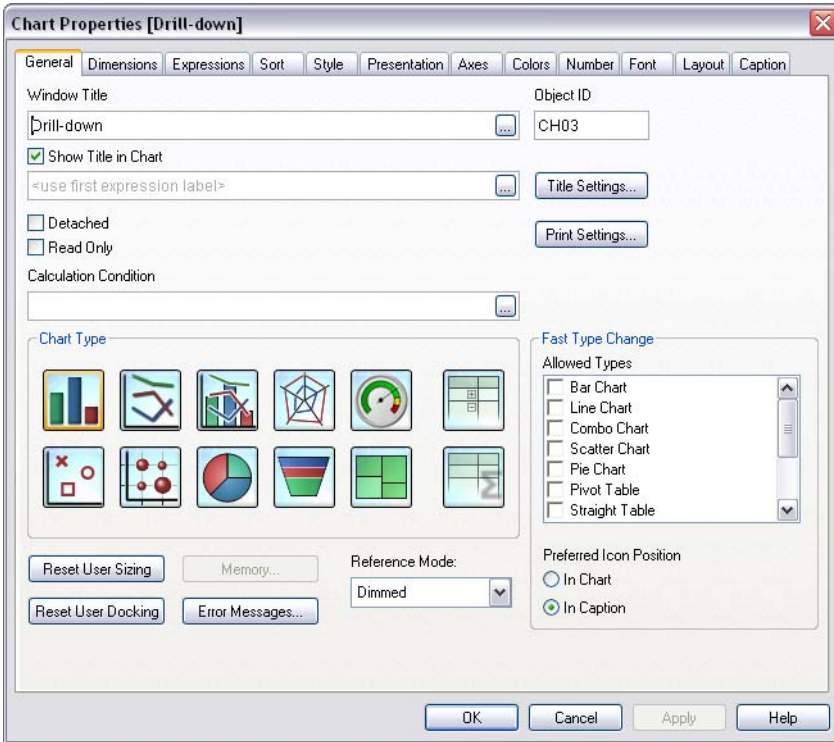


Figure 1. The General page in the Chart Properties dialog

The **General** page, where titles and chart type can be set, is the first page of the chart wizard and the properties dialog.

Window Title

The title to be displayed in the window header. The title may be defined as a calculated label expression (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Show Title in Chart

By default, the label of the first expression defined is set as chart title. Clear the check box if you do not wish any chart title to appear. To get back to the original title, simply mark the check box. The title may be defined as a calculated formula (see page 467) for dynamic update. To have the text on multiple lines: press ENTER. The chart title is not shown in pivot tables and straight tables.

Title Settings...

Opens the **Title Settings** dialog (see page 25 for details), where advanced settings can be made for the chart title.

Object ID

The unique ID of the current chart. Upon creation, every QlikView layout entity is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of entity, as well as a number. The first chart of a document will be assigned the ID CH01. An object ID may be changed to any other string that is currently not used for any other sheet object, sheet, bookmark, report or alert in the document.

Detached

Marking this check box will detach the chart, so that it will no longer be dynamically updated when selections are made.

Read Only

By marking this check box the chart becomes read only, i.e. selections cannot be made by clicking or painting with the mouse in the chart.

Calculation Condition

Typing an expression in this text box sets a condition to be fulfilled for the chart to be calculated. If the condition is not fulfilled, the text 'Calculation condition unfulfilled' will be displayed in the chart. By clicking the ... button the **Edit Expression** dialog is opened for easier editing of long formulas.

Chart Type

In this group you select what type of chart to use: bar chart, line chart, combo chart, scatter chart, pie chart, radar chart, funnel chart, grid chart, block chart, gauge chart, pivot table or straight table.

Bar Chart

The most basic chart type. Displays values as bars of different height.

Line Chart

Instead of being displayed in bars, the data can be presented as lines between value points, as value points only or as both lines and value points. Line charts are useful when you want to show changes or trends.

Combo Chart

This option allows you to combine the features of the bar chart with those of the line chart: you can show the values

of one expression as bars while displaying those of a second expression as lines or symbols.

Radar Chart

The radar chart is a form of the line chart where the x-axis is wrapped around 360 degrees. The result is similar to a spider web or a radar screen.

Scatter Chart

The scatter chart presents pairs of values from two expressions. The expression values are reflected on the two axes. This chart type is useful when you want to show data where each instance has two numbers, e.g. country (population and population growth).

Grid Chart

Similar to the scatter chart but plots dimension values on the axes and uses an expression to determine the plot symbol. A special mode makes it possible to show a third dimension in the form of small pie charts as plot symbols.

Pie Chart

Pie charts normally show the relation between a single dimension and a single expression, but can sometimes have two dimensions.

Funnel Chart

The funnel chart is typically used for showing data in flows and processes. From a display standpoint it is related to the pie chart. The chart may be shown with either segment height/width or segment area proportional to data. It is also possible to draw the chart with equal segment heights/widths, disregarding data points.

Block Chart

Block charts show the relation between expression values as blocks of varying area. Up to three dimensions can be shown, where each dimension block is sub-divided in sub-blocks. Often an extra expression is used to calculate the color of each block.

Gauge Chart

Gauge charts are typically used for displaying a single expression value without any dimensions. QlikView offers a wide range of gauge variants.

Pivot Table

Presents dimensions and expressions in table form. Sub-totals can be shown and the data can be grouped as a cross table with several dimensions.

Straight Table

As opposed to the pivot table, the straight table cannot display sub-totals or serve as a cross table. On the other hand, any of its columns can be sorted and each of its rows contains one combination of dimension(s)+expression(s).

Fast Type Change

In the **Fast Type Change** group you can enable an icon in the chart from which the user can change chart type without going through the chart properties dialog. Right-click the icon to see a drop-down list of chart types and select one by a simple click.

Allowed Types

In this list you can select which chart types that should appear in the drop-down list. Two or more types have to be selected for fast type change to be enabled.

Preferred Icon Position

In graphical charts the fast type change icon can be positioned either inside the chart or in the sheet object caption. In table charts the caption is the only alternative. Select **In Chart** if you want the fast change icon to appear inside graphical charts. If you select **In Caption** the icon will always appear in the caption.

Reset User Sizing

By pressing this button all user sizing of legend, title etc. in graphical charts will be reset. Docking of individual items will not be affected.

Reset User Docking

By pressing this button all user docking of legend, title etc. in graphical charts will be reset.

Memory...

Opens the **Calculation Memory Settings** dialog, in which it is possible to increase the memory allocated to the chart. This is only necessary for large and complex charts. The dialog is only available if the **Ignore Calculation Limits** setting has been unchecked in the **Objects** page in **User Preferences**. See **Memory Calculation Settings** on page 125 in Book I.

Error Messages

Opens the **Custom Error Messages** dialog (see page 471), where you can define custom error messages to be displayed instead of the standard ones.

Reference Mode

Here you may choose between some different options for how the reference background should be plotted when using the **Set Reference** option from the chart's context menu. This setting is only meaningful for some types of charts.

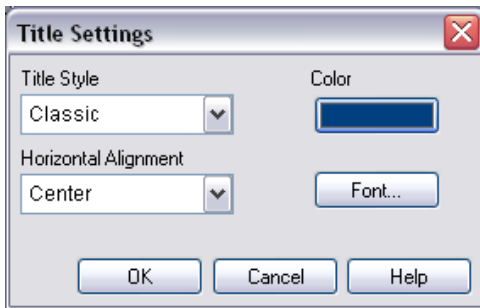
Title Settings dialog

Figure 2. The Title Settings dialog

The Chart title can be formatted in a number of ways using this dialog.

Title Style

Sets the basic style for the title. You may choose from:

Classic

Title area without border and with transparent background.

Divider

Title area with transparent background. Line divider between title and plot area. The divider line is drawn using the color specified under **Color**.

Glass

Title area with semi-transparent glass-like surface as background.

Tinted

Title area with semi-transparent colored surface as background. Uses the color specified under **Color**.

Solid

Title area with solid colored surface as background. Uses the color specified under **Color**.

Background Color

Sets the color of the title background for selected styles. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Horizontal Alignment

Specifies how the title text is positioned in relation to the plot area. The alternatives are **Left**, **Center** and **Right**.

Font...

Sets the font to be used in the chart title. The standard **Font** dialog for charts opens when you click the button.

2 BAR CHART

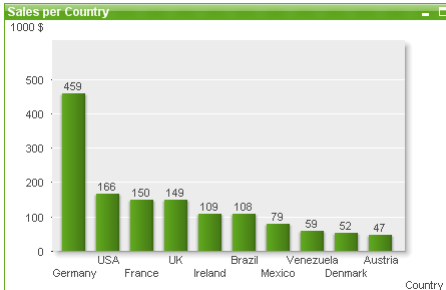


Figure 3. The bar chart is the most basic chart type.

2.1 Creation

Bar charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens **chart properties** dialog that will help you create the chart.

If you want to create a simple bar chart and find the great number of different settings available overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu (**QuickChart**), or click the **QuickChart** button in the toolbar. The wizard appearing will help you to create a chart in a quick and easy way. For more information, see page 265.

2.2 Bar Chart Object Menu

The bar chart **Object** menu is found as the **Object** menu when a bar chart is active. It can also be opened as a context menu by right-clicking on a bar chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the charts static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Set Reference

By choosing this option you set a chart reference, i.e. a fixed plot of the chart with the current selections. When further selections are made in the document the reference plot remains, dimmed in the background. Chart axes etc. will be adjusted to always include the maximum of the background data set and the current data set. The current data set is always plotted on top of the reference plot, i.e. some portions of the reference plot may be obscured by the current data set plot. The way the background is dimmed can be controlled via the **Reference Mode** setting on the **General** page of the **Chart Properties** dialog. The display of reference chart plots is only possible in some chart types, e.g. bar charts, line charts, combo charts, radar charts, scatter charts, grid charts and gauge charts with needles. It is not possible to set reference for a chart that contains a drill-down or a cyclic group. The reference will be lost when closing the document or when reloading data.

Clear Reference

This menu item replaces the **Set Reference** menu item when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Deselects the values of all the fields in the bar chart.

Print...

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from QlikTech's homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the chart.

2.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **chart properties** dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

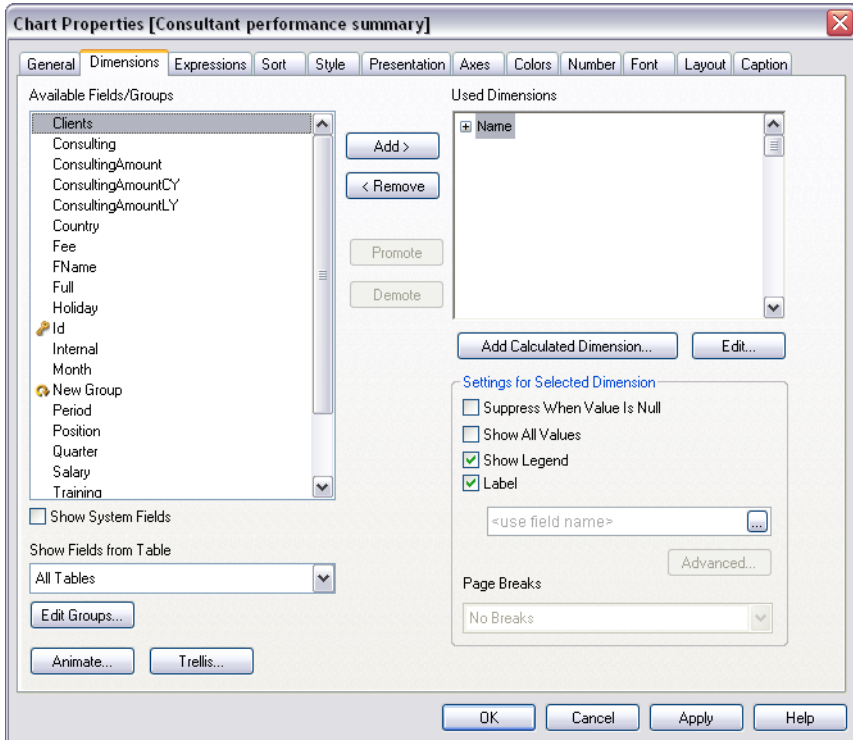


Figure 4. The Dimensions page in the Chart Properties dialog

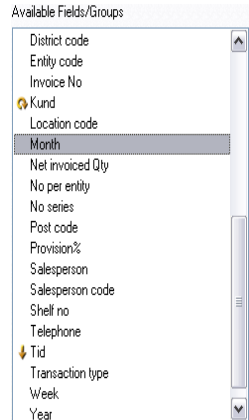
In the **Dimensions** page, the dimensions to be shown in the chart can be set.

Two things are important to think about when making charts:

- What do you want to look at? What should the sizes of the bars in the bar chart correspond to? The answer might be the Sum of Sales, or something similar. This is set on the **Expressions** page.
- What do you want to group it by? Which field values do you want to use as labels for the bars in the bar chart? The answer might be per Country, or something similar. This is set on the **Dimensions** page.

Available Fields/Groups

A list of the fields and field groups that can be used as dimensions (e.g. on the x-axis in a standard bar chart). Names of field groups are preceded by symbols: a straight arrow, indicating a drill-down field group, or a curved arrow, indicating a cyclic field group. Groups are defined in the **Document Properties** dialog (**Groups** page). For further information on drill-down and cyclic field groups, see page 469. All fields which appear in more than one input table will be preceded with a key symbol. Pie charts, line charts and scatter charts cannot display more than two dimensions. In bar charts, up to three dimensions can be shown. To display a field or field group in the chart, select it and click the **Add >** button to move it to the column of **Fields/Groups Used as Dimensions**.



Show Systems Fields

Check this box to make the system fields visible in the **Available Fields/Groups** column.

Show Fields from Table

Normally this drop-down shows the value **-All Tables-** and the **Available Fields/Groups** list contains all fields and groups of the document. By selecting a specific table name in the drop-down list you may limit the **Available Fields/Groups** list to only fields from that table. Finally you may select **-All Tables (Qualified)-** which will show all fields in the document qualified with their table name. A field will appear one time for each table it appears in.

Edit Groups

Opens the **Groups** page (see page 48) in the **Document Properties** dialog, where groups can be created or edited.

Animate...

Opens the **Animation** dialog (see page 36) where you can use the chart's first dimension for animation. Animation is only available for bitmap charts excluding pie charts. Some functionality limitations apply when using animation.

Trellis...

Opens the **Trellis Settings** dialog (see page 38) where you can create an array of charts based on the first dimension.

Used Dimensions

This list contains the dimensions currently selected for use in the chart. The number of dimensions that can be used varies with the type of chart. Superfluous dimensions for any given type will be disregarded. When used in tables, the dimension data cells can be dynamically formatted by means of attribute expressions. Whenever an attribute expression is entered for a dimension, its icon will turn from gray scale to color, or as in the case of **Text Format** from gray to black. These settings will have precedence over chart settings. Click on the “+” expansion icon in front of any dimension to display the placeholders or the dimension’s attribute expressions. They are:

Background color

Double click on **Background Color** in order to enter an attribute expression for calculating the cell background of the dimension cell. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used.

Text Color

Double click on **Text Color** in order to enter an attribute expression for calculating the cell text color of the dimension cell. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used.

Text Format

Double click on **Text Format** in order to enter an attribute expression for calculating the font style of text in the table cell for each dimension cell. The expression used as text format expression should return a string containing a **B** for bold text, **I** for italic text and/or **U** for underlined text.

Add

By clicking this button, you move selected field(s) and field group(s) from the column of available fields/groups to the column of **Used Dimensions**.

Remove

By clicking this button, you move selected field(s) and field group(s) from the column of displayed fields/groups to the column of available fields/groups.

Promote

Promotes the selected field or field group in the **Used Dimensions** list. Use this button if you want to change the order of the dimensions.

Demote

Demotes the selected field or field group in the **Used Dimensions** list. Use this button if you want to change the order of the dimensions.

Add Calculated Dimension...

Adds a new dimension and opens it for editing in the **Edit Expression** dialog (see page 249 for details). A chart dimension is often a single field, but can also be dynamically calculated. A calculated dimension consists of an expression involving one or more fields. All standard functions may be used. Aggregation functions may not be used but the **aggr** function can be included for achieving nested aggregation. Calculated dimensions will only work in QlikView versions 7.5 and later.

Edit...

Opens the dimension for editing in the **Edit Expression** dialog. See **Add Calculated Dimension...** above for details on calculated dimensions.

Settings for Selected Dimension

In this group you find settings for individual dimensions.

Suppress When Value Is NULL

If this check box is marked, the field selected in the **Used Dimensions** above will not be displayed in the chart if its value is NULL.

Show All Values

Select this box to show all the dimension field values in the chart, including logically excluded values.

Show Legend

Mark this check box to show legend (names of field values) on the x-axis for the selected dimension field.

Label

Mark this check box to show a dimension label (the field name) on the x-axis. The label may be defined as a calcu-

lated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Advanced

Opens the **Advanced Field Settings** dialog (see page 149).

Page Breaks

Here you may choose to insert page breaks when printing a table wherever the value of the current dimension changes. There are three modes:

No Breaks

Page breaks are inserted automatically where needed without any relation to the data.

Forced Breaks

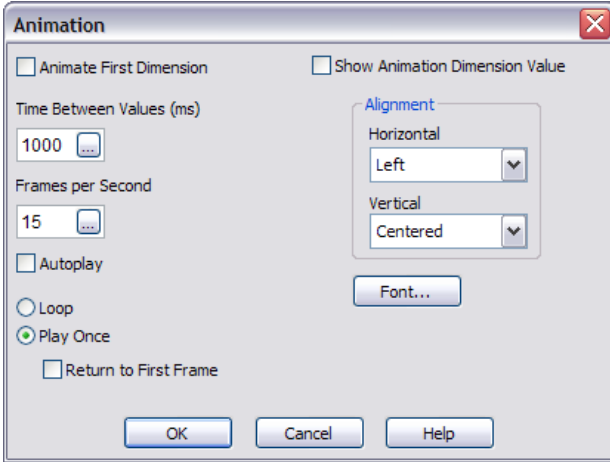
A page break is always inserted when the dimension changes value.

Conditional Breaks

No page break will be inserted if all rows with the following dimension value can be printed on the current page.

Note Page breaks are only visible when printing and not in the layout on the screen.

Animation dialog

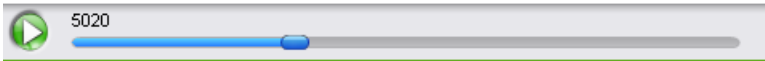


Animate First Dimension

By marking this check box you indicate that the chart's first dimension should be used for chart animation. Animation is only available for bitmap charts excluding pie charts. Some functionality limitations apply when using animation. For example, it is not possible to make painted or clicked selections in an animated chart. Trend lines will not be drawn in animated charts. Animation will only be possible when there is more than one possible value in the animation dimension.

When a chart is animated an **Animation Bar** will appear at the bottom of the chart plot area. The **Animation Bar** features a **Play** button, which starts the animation. When the animation is running the **Play** button is replaced by a **Pause** button. You may stop or start the animation whenever you like using these controls. A **Progress Bar** shows the progress of the animation. You may animate manually by pointing at the progress bar handle with the mouse, pressing down the left mouse button and dragging it to any position. Manual animation normally skips interpolated frames (see **Frames Per Second** setting below) and moves only between actual values in the animation dimension. By pressing the ctrl key while dragging it is possible to drag over interpolated frames. Above the progress bar the value of the animation dimension for the current frame (in case

of interpolated frames the previous actual animation dimension value) will be shown.



Time Between Values (ms)

Sets the time in milliseconds between each value in the animation dimension. This value may be given as a calculated expression (see page 467).

Frames per Second

Sets the number of frames per second. QlikView will interpolate plotting between the actual values of the animation dimension. The value must be an integer between 1 and 30. This value may be given as a calculated expression (see page 467)

Autoplay

Mark this check box if you want an animation to start automatically whenever a selection is made in the document.

Loop

Select this option if you want the animation to play repeatedly until stopped with the **Pause** button in the **Animation Bar**.

Play Once

Select this option if you want the animation to run only once from beginning to end, whenever it is started.

Return to First Frame

Mark this check box if you want the animation to return to the first frame after completion.

Show Animation Dimension Value

By marking this check box the value of the data will be shown in the chart during the animation.

Alignment

Choose the **Horizontal** and **Vertical** alignment of the displayed values.

Font

Sets the font for the displayed value.

Trellis Settings

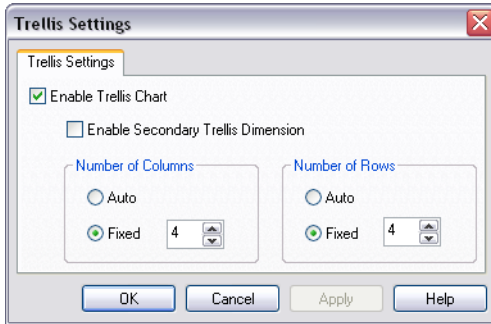


Figure 5. The Trellis Settings dialog

Enable Trellis Chart

Mark this check box to create an array of charts based on the chart's first dimension.

Enable Secondary Trellis Dimension

Mark this check box to include the second dimension in the trellis chart. If a secondary dimension is used, the first dimension values will be displayed as columns in the trellis matrix, whereas the second dimension values will be displayed as rows in the trellis matrix.

Number of Columns

Choose **Auto** to let QlikView decide how many columns to display or choose **Fixed** to set the number yourself.

Number of Rows

Choose **Auto** to let QlikView set the number of rows to display or choose **Fixed** to set the number yourself

Expressions

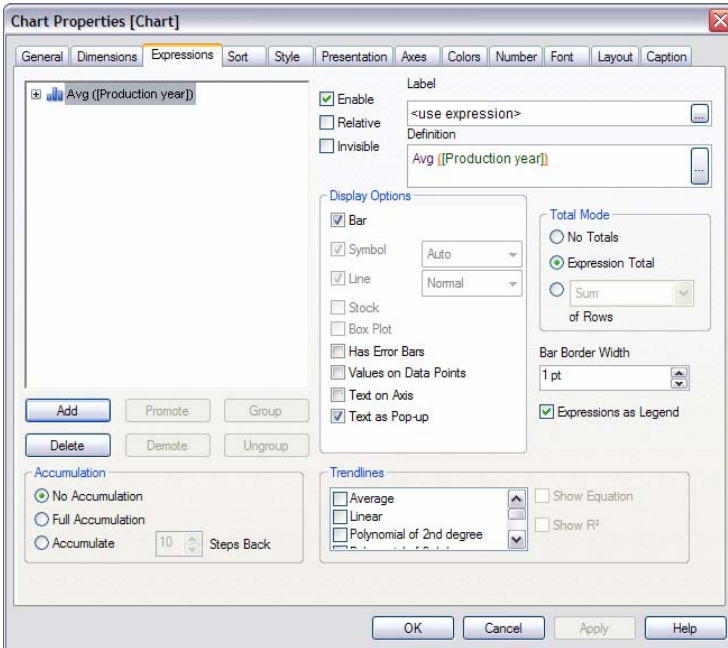




Figure 6. The Expressions page in the Chart Properties dialog

On the **Expressions** page you define expressions to be displayed in the chart.

Expressions



The upper left corner of the dialog contains a list of the chart's expressions. An expression defines the calculated contents of a chart, e.g. the data shown on the y-axis in an expression column.

Navigating the tree controls:

The expression list is in fact a tree control with a large set of control options. In front of each expression you will find an expansion icon . By clicking on the expansion icon you open up underlying sub-expressions or attribute expressions. The icon will then shift to a collapse icon . By clicking that icon you collapse the display of any subexpressions or attribute expressions. In complex cases up to three levels of expansion may be possible.

Display options / plot modes:

After the expansion/collapse icon on each row you will see an icon for each available attribute expression. These are normally set in the


Display Options group (see below). There is for example one icon for bar , one for line  and so on. In certain chart types it is possible for one expression to be shown in multiple ways, which will be reflected by a list of several icons. The icons shown are dependent on the type of chart chosen. You may find that the icons shown next to a given expression will change if you change the chart from e.g. a bar chart to a combo chart or to a straight table. Only icons reflecting plot types possible in the active chart type will be shown.

Sub expressions:

Certain plot options make use of sub expressions, i.e. a set of two or more expressions that together define the plot symbol. By selecting Stock as Display Option you will create four sub expressions, of which at least the first two must be defined. For Box Plot there will be five. In these two cases the base expression will just be an empty placeholder in the tree control. All expression definitions lie within the sub expressions.

Another instance of sub expressions is when error bars are used. The error bars are defined by one or two sub expressions. In this case however the base expression contains its own definition.

Attribute expressions:

Expression data can be dynamically formatted by means of attribute expressions. Click on the ‘’ expansion icon in front of any expression to display the placeholders for the dimension’s attribute expressions. You may choose to use any number and combination of attribute expressions on a given base expression. Whenever an attribute expression is entered for a dimension, its icon will turn from gray scale to color, or as in the case of **Text Format** from gray to black. Formatting defined by means of attribute expression supersede default formatting of the data plots in terms of color etc. The available attribute expression types are:

 **Background color**

Click on **Background Color** in order to define an attribute expression for calculating the basic plot color of the data point. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic) which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used.

A Text Color

Click on **Text Color** in order to define an attribute expression for calculating the text color of text associated with the data point. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used.

T Text Format

Click on **Text Format** in order to enter an attribute expression for calculating the font style for text associated with the data point. The expression used as text format expression should return a string containing a B for bold text, I for Italic text and/or U for underlined text.

Pie Popout

Only applicable on pie charts. Click on **Pie Popout** in order to enter an attribute expression for calculating whether the pie slice associated with the data point should be drawn in an extracted “popout” position.

Bar Offset

Only applicable on bar charts. Click on **Bar Offset** in order to enter an attribute expression for calculating an offset for the bar or bar segment accordingly. This is useful e.g. when making so called waterfall charts.

Line Style

Only applicable on line, combo and radar charts. Click on **Line Style** in order to enter an attribute expression for calculating the line style for the line or line segment associated with the data point.

The relative width of the line can be controlled by including a tag `<Wn>` where n is a multiplying factor to be applied on the default line width of the chart. The number n must be a real number between 0.5 and 8. Example: `<W2.5>`.

The style of the line can be controlled by including a tag `<Sn>` where n is an integer between 1 and 4 indicating the style to be used (1=continuous, 2= dashed, 3=dotted, 4=dashed/dotted). Example: `<S3>`.

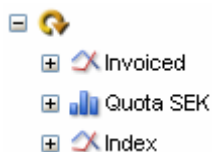
The <Wn> and <Sn> tags can be freely combined, but only the first occurrence of each counts. The tags must be enclosed by single quotations.


Show Value

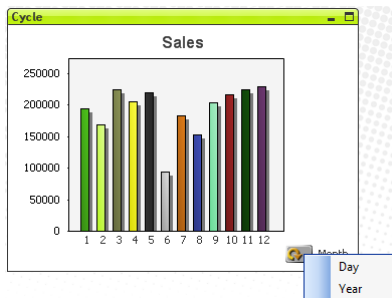
Only applicable on bar, line and combo charts. Click on **Show Value** in order to enter an attribute expression for calculation whether the data point plot should be complemented with a “number on data point value“, even if **Values on Data Points** has not been selected for the main expression. If **Values on Data Points** is selected for the main expression the attribute expression will be disregarded.

Expression cycle groups:

Finally there is the concept of expression groups. Any number of base expressions (with their respective sub expressions and attribute expressions) may be bundled together in a group. A group is shown in the



list as a placeholder with a  cycle icon. By expanding the group, you can see the expressions it contains. In the chart only the first expression in the list will be used for display. The expression to be shown out of the group can be changed by clicking the cycle icon in the chart or in the table expression column header. By right-clicking the cycle icon, you get a pop-up list of the available expressions for direct selection. When you cycle the group the second expression will be promoted to the top of the list and the previously active expression demoted to the bottom of the list.



You can add an expression to a group in two ways. The first method is by simply dragging an expression on top of another expression in the list (creates a new cycle group). The second method is by select-

ing an expression in the list and then click on the **Group** button (see below). The expression will then be grouped with the expression or cycle group residing above in the list.

You can correspondingly remove an expression from a group by dragging it out of the group in the list or by selecting it and pressing the **Ungroup** button (see below). If only one expression remains in the cycle group after the extraction, that expression will also be extracted and the cycle group will cease to exist.

Add

By clicking this button, you get to the **Edit Expression** dialog (page 249), in which it is possible to create new expressions. This option is also available in the context menu that appears when right-clicking on an expression in the list of expressions.

Delete

Deletes the selected expression.. The option is also available in the context menu that appears when right-clicking on an expression in the list of expressions.

Copy

The option is only available in the context menu that appears when right-clicking on an expression in the list of expressions. When using this command on a main expression, all data and settings associated with the expression (including label) will be copied to the clipboard as a piece of xml. The expression may then be pasted back into the same chart or into any other QlikView chart in the same or another document. If you use the command on an attribute expression, only the attribute expression definition will be copied. An attribute expression may then be pasted onto any main expression in the same or another chart.

Export...

The option is only available in the context menu that appears when right-clicking on a main expression in the list of expressions. When using this command on a main expression, all data and settings associated with the expression (including label) may be exported to an xml file. The expression may then be imported back into the same chart or into any other QlikView chart in the same or another document. The command opens the **Export Expression** dialog from which you can choose the destination of the export file. The file will receive the extension Ex.xml.

Paste

The option is only available in the context menu that appears when right-clicking in the list of expressions and only if an expression has previously been copied to the clipboard. If a main expression has previously been copied to the clipboard, you may paste it into the blank area in the list of expressions, creating a new expression identical to the copied one. If an attribute expression has been copied, you may paste it onto a main expression.

Import...

The option is only available in the context menu that appears when right-clicking in the blank area in the list of expressions. The command opens a dialog where you can browse to previously exported expressions. The imported expression will appear as a new main expression in the chart.

Promote

Promotes the selected expression one step up in the list. Use this button if you want to change the order of the expressions.

Demote

Demotes the selected expression one step down in the list. Use this button if you want to change the order of the expressions.

Group

It is possible to group two or more expressions, so that only one of them is displayed in the chart at any given time. The expression to be shown out of the group can be changed by clicking the cycle icon in the chart or in the table expression column header. By right-clicking the cycle icon, you get a pop-up list of the available expressions for direct selection (see the picture above). You may assign an expression to a cycle group by selecting an expression in the list and then click on the **Group button**. The expression will then be grouped with the expression or cycle group residing above it in the list.

Ungroup

This command is only available when an expression belonging to a cycle group has been selected in the **Expressions** list above. By clicking on the button, the expression will be moved out of the cycle group and placed as an independent expression after the cycle group in the list. If only one expression remains in the cycle group after the extraction, that expression will also be extracted and the cycle group will cease to exist.

Enable

Marking this check box will enable the selected expression. If the box is not marked, the expression will not be used at all.

Relative

Marking this check box will set the chart to show the result in percent instead of absolute numbers. This option is not available for pivot tables.

Invisible

Marking this check box will cause the expression not to be plotted while maintaining the space normally allocated for its plotting.

Label

The label of the expression. Enter the name you wish to display in the chart. If no text is entered, the label will be automatically set to the expression text.

Definition

Shows the composition of the selected expression. It is possible to edit the expression directly in this box. By clicking the ... button the full **Edit Expression** dialog is opened.

Display Options

In this group you can select how the data points are plotted. Some options are only available for certain chart types. Some options cannot be combined and some options will utilize one or more additional expressions in order to create complex plots.

Bar

Shows the values of the selected expression as bars. This option is only available for bar and combo charts.


Symbol

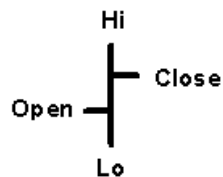
Shows the values of the selected expression as symbols. This option is only available for line and combo charts. Choose between several different symbols in the drop-down menu.

Line

Shows the values of the selected expression as a line. This option is only available for line and combo charts. Choose between **Normal**, **Smooth** and three different **Plateau** lines in the drop-down menu.


Stock

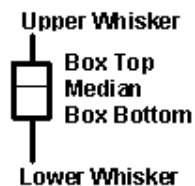
Mark this check box to plot the expression as a stock marker. The expression will be preceded by the icon  in the **Expressions** list and appear as an empty placeholder with four sub expressions.



The first sub expression will be used for plotting the **Hi** point of the stock marker. Sub expression two will be used for the **Lo** point. These two expressions must contain valid definitions in order for the stock marker to be drawn. Sub expression three is optional but is otherwise used for the **Close** point of the stock marker. Sub expression four is also optional but is otherwise used for the **Open** point of the stock marker. New empty sub expressions will be created automatically when **Stock** is marked for a base expression. When **Stock** has been selected for an expression you cannot select **Bar**, **Line**, **Symbol**, **Box Plot** or **Has Error Bars** for the same expression. **Stock** cannot be selected for an expression if any of those options are already selected for that expression. Stock plot is only available in combo charts.

Box Plot




Mark this check box to plot the expression as a box plot, often used for display of statistical data. The expression will be preceded by the icon  in the **Expressions** list and appear as an empty placeholder with



five sub expressions. The first sub expression will be used for plotting the **Box Top** point of the box plot. Sub expression two will be used for the **Box Bottom** point. These two expressions must contain valid definitions in order for the box plot to be drawn. Sub expression three to five are optional. If used, sub expression three defined the **Median**, sub expression four the **Upper Whisker** and sub expression five the **Lower Whisker**. A common extension to the box plot are so called outliers for extreme values. These can be achieved by plotting separate expressions as symbol. New empty sub expressions will be created automatically when **Box Plot** is marked for a base expression. When **Box Plot**

has been selected for an expression you cannot select **Bar**, **Line**, **Symbol**, **Stock** or **Has Error Bars** for the same expression. **Box Plot** cannot be selected for an expression if any of those options are already selected for that expression. Box plot is only available in combo charts.

Has Error Bars

Mark this check box to utilize one or two expressions following the selected expression as auxiliary expressions for error bars plotted on top of the main expression's data points. If **Symmetric** is selected only one auxiliary expression will be used and plotted symmetrically around the data point. If **Asymmetric** is selected two auxiliary expressions will be used and plotted above and below the data point respectively. The error bar expressions should return positive numbers. The auxiliary expressions utilized for error bars are preceded by the icons  (symmetric),  (asymmetric high) or  (asymmetric low) in the **Expressions** list and cannot be utilized for anything else in the chart. If there are no expressions already defined after the selected expression, new dummy auxiliary expressions will be created automatically. This option is only available in bar, line and combo charts.

Values on Data Point

Mark this check box to have the result of the expression plotted as text on top of the data points. This option is only available in bar, line, combo and pie charts. When used for pie charts, the value will be shown next to the pie slices.

Text on Axis

Mark this check box to have the result of the expression plotted as text at each x-axis value, the axis and the axis labels. This option is only available in bar, line and combo charts.

Text as Pop-up

Mark this check box to have the result of the expression shown in the pop-up balloon messages appearing when hovering over a data point in a chart in the layout. This option can be used with or without any of the other display options. It is thus possible to have an expression that does not appear in the chart itself but only in hover pop-ups.

Accumulation

By choosing between the settings in this group, you decide whether the values in the chart should be accumulated or not. In an accumulated chart, each y-value is added to the y-value of the following x-value. In an accumulated bar chart showing the sum of sales per year, e.g., the value of the year 1996 is added to that of 1997. If your chart contains several expressions, select the expression (in the **Expressions** box) whose values you wish to accumulate. Accumulation is not available for pivot tables.

No Accumulation

If this check box is marked, the y-values of the selected chart expression will not be accumulated.

Full Accumulation

Marking this box will result in each y-value accumulating all previous y-values of the expression. See above under **Accumulation**.

Accumulate N steps back

By entering a number in the box, you set the number of y-values in the expression to be accumulated. See above under **Accumulation**.

Total Mode

By selecting one of the options in this group, you decide how the total of the selected expression is to be calculated. This setting is important for relative display or when showing totals.

No Totals

If this option is selected no total will be calculated for the expression.

Expression Total

If this option is selected, the expression total will be calculated using all the values of the field. If, e.g., the selected column contains the average salaries for different business categories, choosing the **Expression Total** option will result in the average salary for all the business categories.

F(x) of Rows

If this option is selected, the individual values of each data point (each bar in a bar chart, each row in a straight table etc.) for the selected expression will be summed up, aggregated, using the selected aggregation function (typically summed up). This option is not available for pivot tables.

Bar Border Width

Specifies the width of the border line around bars in bar and combo charts. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Expressions as Legend

When several expressions are used, it is possible to display a legend showing the expressions and the corresponding colors.

Trend lines

In selected QlikView charts expression plots can be complemented or replaced by statistical trend lines. Trend lines can only be displayed in scatter charts, line charts and in bar/combo charts with maximally one dimension and one expression shown as bars. For other types of charts, the settings in the **Trend lines** box are unavailable and have no effect. In scatter charts the data points are treated as if $y=f(x)$. For bar, line and combo charts it is allowed to deselect all options under **Plot Options** and still add trend lines, which will then be plotted without the underlying data points. Trend lines in bar, line and combo charts may be extrapolated by specifying a forecast and/or backcast interval (**Axes** page). The extrapolated lines will be dotted. Trend lines in charts with a discrete x-axis will be shown as lines with symbols. On a continuous axis only the line will be shown.

Average

The average is plotted as a straight line.

Linear

A linear regression line is plotted.

Polynomial of 2nd degree

A polynomial trend line of the second degree is plotted.

Polynomial of 3rd degree

A polynomial trend line of the third degree is plotted.

Polynomial of 4th degree

A polynomial trend line of the fourth degree is plotted.

Exponential

An exponential trend line is plotted.

Show Equation

If this check box is marked for a specific expression, the expression's trend lines will be complemented by the trend-line equation expressed as text in the chart.

Show R²

If this check box is marked for a specific expression, the expression's trend lines will be complemented by the coefficient of determination expressed as text in the chart.

Sort

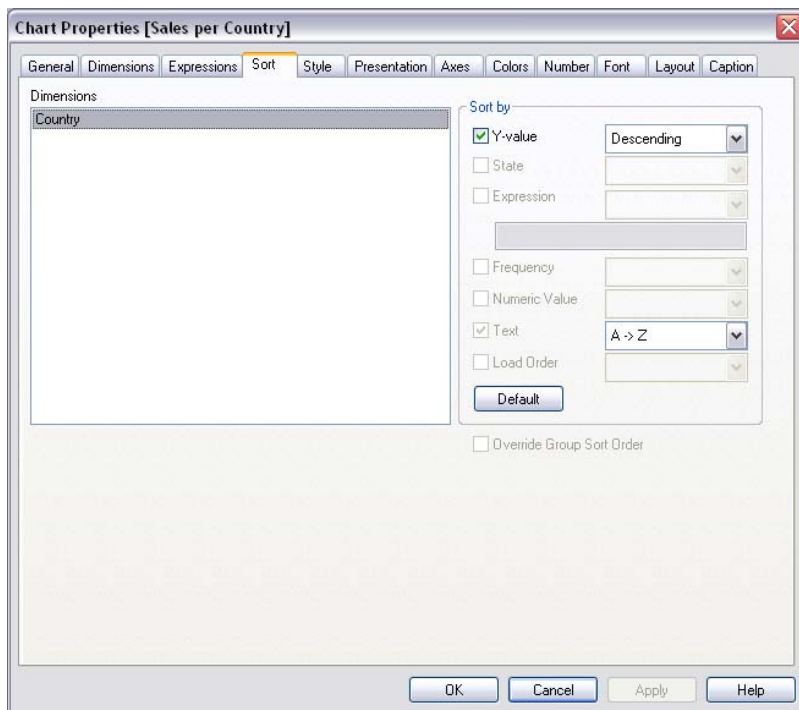


Figure 7. The Sort page in the Chart Properties dialog

On this page you can sort the values of the dimensions according to one of the available sort orders.

Dimensions

A list of the fields you have chosen as chart dimensions on the **Dimensions** page. The field marked is the one whose sort order you are currently changing.

Sort by

In the **Sort by** group, the sort order of the dimension values (bars in the bar chart, slices in the pie chart and usually rows in the tables)

can be set. Pivot tables can only be sorted by some property of the first column.

Y-value

Here it is possible to set whether the dimension values should be sorted according to the numeric value of the y-axis. This option is not available for calculated dimensions.

State

Sorts the values according to their logical state (selected, optional or excluded).

Expression

Sorts the values according to the expression entered into the text edit box below this sort option. This option is not available for calculated dimensions.

Frequency

Sorts the values by frequency (number of occurrences in the table).

Numeric Value

Sorts the values according to their numeric value.

Text

Sorts the values in alphabetical order.

Load Order

Sorts the values according to their initial load order.

Default

Sets the default sort order.

Override Group Sort Order

This check box is only available when a group dimension is selected in the **Dimensions** list. Normally the sort order of a group dimension is determined for each field in a group via the group properties. By marking this check box you can override any such settings on group level and apply a single sort order for the dimension, regardless of which field is active in the group.

The order of priority is **Y-value**, **State**, **Expression**, **Frequency**, **Numeric Value**, **Text** and **Load Order**. Each of these sort criteria can be set to ascending or descending.

Style

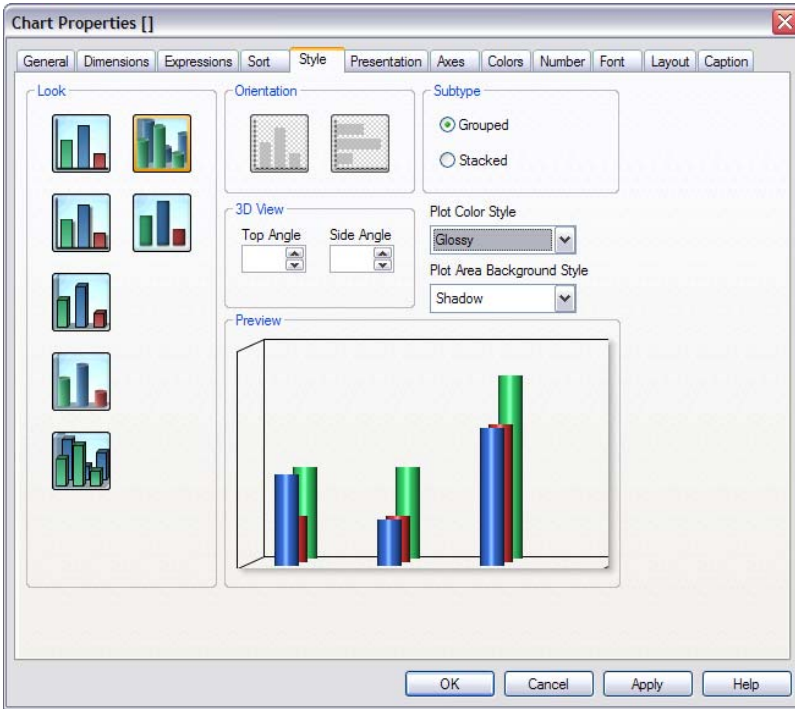


Figure 8. The Style page for bar charts in the Chart Properties dialog

On this page you select the basic style for the bar chart.

Look

Select one of the available styles.

Orientation

Here you can set the orientation of the chart to either vertical or horizontal as indicated by the icons.

Chart Subtype

In this group it is possible to choose between **Grouped** and **Stacked** mode, provided that the chart has exactly two dimensions or exactly one dimension but more than one expression. Negative values in stacked bars are stacked separately downwards below the x-axis.

For the presentation of bar charts with multiple dimensions and expressions the following principles apply:

- A maximum of two dimensions can be shown on the x-axis.

- A third dimension can be shown with multicolored stacked bars. Only table charts can display more than three dimensions.
- When two or more expressions are enabled, the two first dimensions are shown on the x-axis and the expression with multicolored stacked bars.

Dimension	Expression	Subtype
1	1	Single bar
1	2 or more	Expressions are grouped or stacked
2	1	Dimensions are grouped or stacked
2	2 or more	Dimensions are grouped
3	1	1st and 2nd dimensions are grouped, 3rd dimension is stacked
3	2 or more	1st and 2nd dimensions are grouped, expressions are stacked
4	1	1st and 2nd dimensions are grouped, 3rd dimension is stacked
4	2 or more	1st and 2nd dimensions are grouped, expressions are stacked.

3D View

The settings in this group define the angle from which the chart is viewed in 3D modes.

Top Angle

Defines the vertical angle of the 3D view. The value must be an integer between 0 and 30.

Side Angle

Defines the side angle of the 3D view. The value must be an integer between 0 and 45.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors** page will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all bars.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

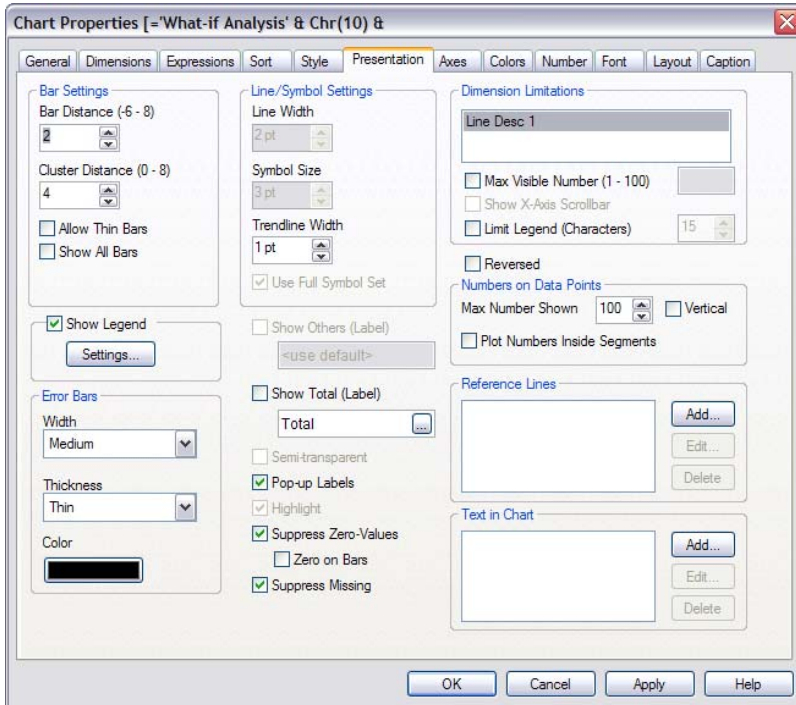


Figure 9. The Presentation page for bar charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the bar chart is presented on the screen.

Bar Settings

In the **Bar Settings** group you can set display options for bars in the chart.

Bar Distance

By entering an appropriate number, it is possible to set the distance between the bars in the cluster. It is also possible to enter negative numbers. Values between -6 and 8 are allowed.

Cluster Distance

By entering an appropriate number, it is possible to set the distance between clusters in the chart. Values between 0 and 8 are allowed.

Allow Thin Bars

For charts with non-continuous x-axis, QlikView will only display as many data points as can be accommodated in the plot area available. Remaining data points are truncated from the chart. By default bars will be drawn with a minimum width of four pixels, so that they are clearly distinguishable. Mark this check box to allow bar width down to one pixel.

Show All Bars

For charts with non-continuous x-axis, QlikView will only display as many data points as can be accommodated in the plot area available. Remaining data points are truncated from the chart. Mark this check box to force plotting of all bars. The bars will be compressed and some data points may become obscured by others.

Show Others (Label)

Mark this check box to display a separate x-axis value for “Others”, i.e. an expression value for all dimension values excluded by a limitation under **Dimension Limitations-Max Number**. The 'Others' value will only be meaningful for expressions where **F(x) of Rows** total is meaningful (see “**Expressions**” page 39) and the value will be calculated using the function specified under that option. A label for the 'Others' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Others** label found in the **Objects** page of the **User Preferences** will be used (see page 107 in Book I).

Semi-transparent

By selecting this check box, the chart will be drawn semi-transparent when using area style. Only available to charts with an area style.

Show Total

Mark this check box to display a separate x-axis value for “Total”, i.e. an expression total for all dimension values. The 'Total' value will only be meaningful for expressions where **F(x) of Rows** total is meaningful (see “**Expressions**” page 39) and the value will be calculated using the function specified under that option. A label for the 'Total' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Total** label found in the **Objects**

page of the **User Preferences** will be used (see page 107 in Book I). Only symbol and no line will be plotted for 'Total'.

Pop-up Labels

By selecting this check box the values of the dimension(s) and expression(s) will appear as a pop-up balloon when hovering with the cursor over data point in the chart.

Suppress Zero-Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Zero on Bars

This setting is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Point** is selected for the chart expression under **Display Options** on the **Expressions** page, zero values will appear as text above the data points. In other cases zero values will be suppressed.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Line/Symbol Settings

In the **Line/Symbol Settings** group you can set display options for trend lines in the chart.

Trendline Width

Here you can set the width of the trend lines by entering an appropriate number. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The line can be taken away by setting the width to 0.

Use Full Symbol Set

If this check box is marked, different kinds of symbols available will be used to distinguish data points from each other. If it is not marked, all the symbols will be drawn as plus signs. Not available for bar charts.

Show Legend

Select this check box if you want the legend to be shown in the chart when applicable. If the chart is dimensionless, but has several

expressions, unchecking this box will show the expressions on the axis instead.

Settings...

Opens the **Legend Settings** dialog (see page 60), where advanced settings can be made for the chart legend.

Error Bars

In this group you can set formatting options for error bars, if such are used in the chart.

Width

Specifies the width of the error bars. You may choose between **Narrow**, **Medium** and **Wide**.

Thickness

Specifies the thickness of the error bars. You may choose between **Thin**, **Medium** and **Thick**.

Color

Click the button to select a color for the error bars.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Max Number

Select this check box to limit the number of dimension values to be shown at any one time. Enter the maximum number of values in the scroll box.

Show X-Axis Scroll Bar

Select this check box to show a scroll control in the place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values to be shown. The number of values shown at any one time will be the number set under **Max Number**. This option is only available when **Max Number** is selected.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Reversed

When this check box is marked, the values shown in the chart will be reversed, i.e. if the 10 highest values are shown and the box is checked the chart will then show the 10 lowest values.

Values on Data Points

In the **Values on Data Points** group you can set display options for values on data points, provided that this option has been selected for one or more chart expressions under **Plot Options** in the **Expression** page of the chart's properties.

Vertical

Shows the values vertically.

Max value Shown

In this box you can specify an upper limit for how many data points to show values for in the chart. If no limit is specified, values will be shown for all data points, which may affect the readability of the chart.

Plot Values Inside Segments

Marking this check box will plot values on data points inside the segments instead of on top of them.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 60) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click this button in order to edit its properties in the **Reference Lines** dialog. Double-clicking the reference line in the list has the same effect. See page 60 for details about the **Reference Lines** dialog.

Delete

Mark an existing reference line in the list and click this button in order to delete it from the list.

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog (see page 64) where you may create and edit free-floating texts to be displayed in the chart.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog. Double-clicking the text in the list has the same effect. See page 64 for details about the **Text in Chart** dialog.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Legend Settings dialog

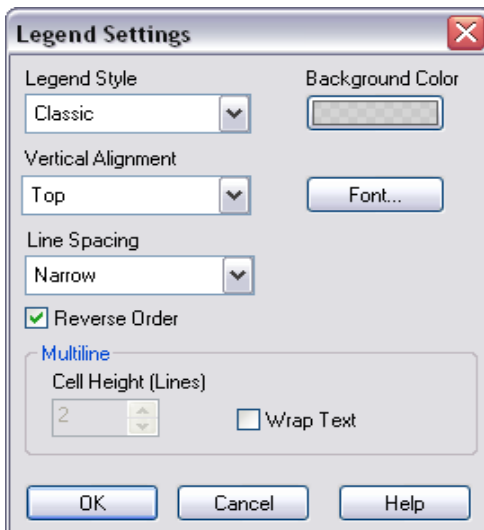


Figure 10. The Legend Settings dialog

The chart legend can be formatted in a number of ways using this dialog.

Legend Style

Sets the basic style for the legend. You may choose from:

Classic

Legend without border and with transparent background.

Glass

Legend with semi-transparent glass-like surface as background.

Frame

Transparent background legend with frame. The frame is drawn using the color specified under **Background Color**.

Solid

Legend with solid colored surface as background. Uses the color specified under **Background Color**.

Tinted

Legend with semi-transparent colored surface as background. Uses the color specified under **Background Color**.

Background Color

Sets the color of the legend background selected styles. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button.

Vertical Alignment

Specifies how the legend is positioned in relation to the plot area, when it needs less vertical space than the plot area. The alternatives are **Top**, **Center** and **Bottom**.

Font...

Sets the font to be used in the chart legend, if other than the default chart font. The standard **Font** dialog for charts opens when you click the button.

Line Spacing

Specifies the distance between items in the legend. You may choose between **Narrow**, **Medium** and **Wide**.

Reverse Order

When marking this check box, the sort order of the legend will be reversed.

Multi Line

In this group you set the options for multiline legend items.

Wrap Text

If this check box is marked, the text of the legend items will be wrapped in two or more lines.

Cell Height (Lines)

If you have selected **Wrap text**, this is where you specify how many lines should be used for each item.

Reference Lines

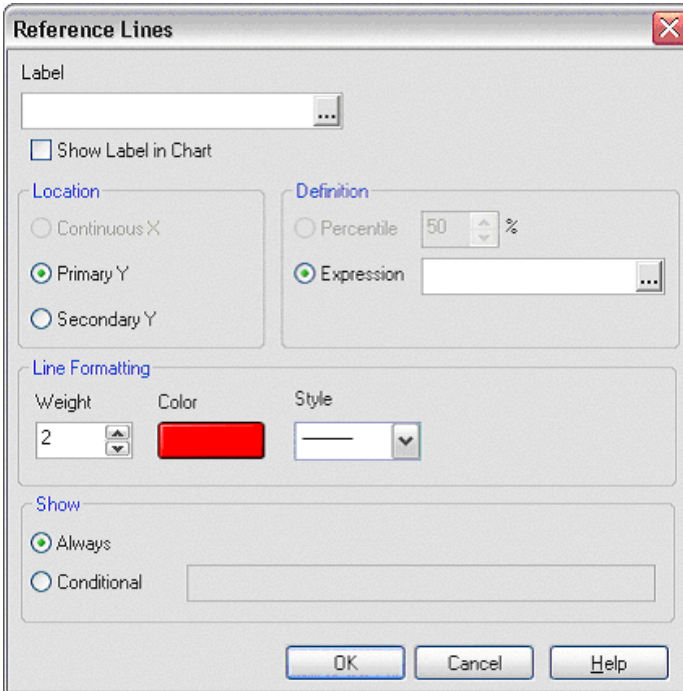


Figure 11. The Reference Lines dialog

A reference line is a line intersecting the chart plot area from a given point on a continuous x- or y-axis. It may be used e.g. to indicate a certain level, point in time etc. in relation to chart data. The reference line is only drawn if it falls within the current range of the axis from which it originates.

Label

Here you may specify a label to be drawn next to the reference line. As default the value of *Expression* will be used.

Show Label in Chart

Mark this check box if you want the label to appear next to the reference line.

Location

In the **Location** group you define from which axis the reference line should originate.

Continuous X

Select this option if you want the reference line to originate from the x-axis. This option is only available if the chart has a Continuous x-axis (see Axes page below).

Primary Y

Select this option if you want the reference line to originate from the primary y-axis (left/bottom).

Secondary Y

Select this option if you want the reference line to originate from the secondary y-axis (right/top).

Definition

The value at which the reference line should be drawn is specified in the **Expression** edit box. The definition could either be a fixed numeric value or a calculated expression (see page 467). By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Line Formatting

In the **Line Formatting** group you define how the reference line will look.

Weight

Specifies the weight of the reference line. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Color

By clicking this button you may select a color for the reference line.

Style

Specifies the style of the reference line, e.g continuous, dashed or dotted.

Show

In this group, it is possible to specify a condition under which the reference line is shown.

Always

The reference line will always be shown.

Conditional

The reference line will be shown or hidden depending on a conditional expression which will be evaluated each time the chart is to be drawn. The reference line will only be visible when the expression returns true.

Chart Texts

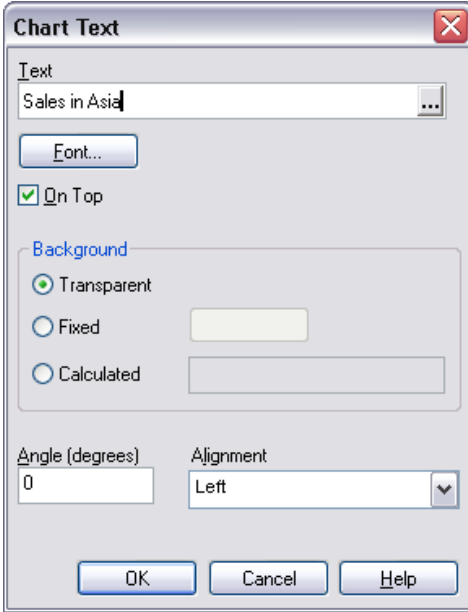


Figure 12. The Chart Texts dialog

Any number of free floating texts may be added to the chart. If a text is placed fully within the plot area, QlikView will attempt to move it relative to the plot area when sizing the chart. If placed at least partly outside the plot area, the text will be moved in relation to the entire chart.

Text

Here you may edit the text currently selected in the list. By clicking the ... button a larger editor window is opened for better editing of long texts or expressions. The text may be defined as a calculated expression.

Font

Sets the font and text color for the text currently selected in the list.

On Top

By marking this check box the text currently selected in the list will be forced to the foreground when drawing the chart.

Background

In the **Background** group you may specify the background as **Transparent**, **Fixed** (select color with the colored button) or **Calculated** (enter the formula for the color in the edit box).

Angle (degrees)

Here you may specify an angle between 0 and 360 degrees for the text. The default is 0.

Alignment

Sets the horizontal alignment of the text within its background to Left, Center or Right.

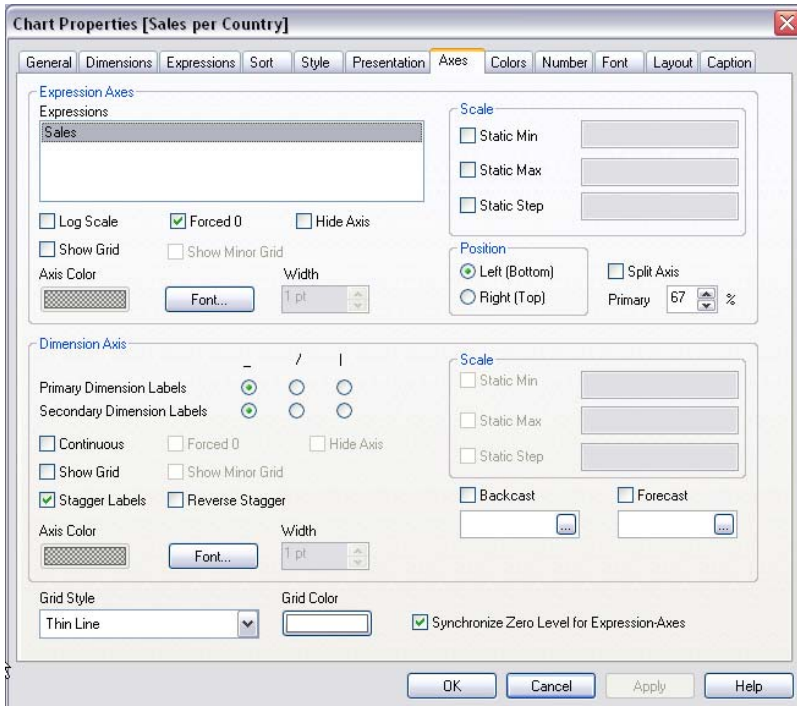
Axes

Figure 13. The Axes page in the Chart Properties dialog

In the **Axes** page, the display properties of the x- and y-axes are set. A chart can have two scales on the y-axis if several expressions are shown. If two scales are used, one will be shown to the left and the other to the right.

Expression Axes

In the **Expression Axes** group you select settings for the y-axes. Every expression has its own axis settings. If two expressions, which are plotted on the same axis have conflicting settings, those of the expression that comes first in order will prevail.

Expressions

In this list you select the expression for which you want to set the axis properties.

Log Scale

Changes the y-axis to a logarithmic scale. A logarithmic scale can only be used when all data points in the chart are positive (>0).

Forced 0

Fixes the lower edge of the chart to zero on the y-axis. This option is not available when a logarithmic axis is used.

Hide Axis

Hides the axis for the selected expression.

Show Grid

Shows a grid in the chart originating from the tick marks of the y-axis. The grid lines can be formatted by using the **Grid Style** and **Grid Color** settings (see below).

Show Minor Grid

Shows minor lines in-between the grid lines. Only available if **Show Grid** is selected. The grid lines can be formatted using the **Grid Style** and **Grid Color** settings (see below).

Axis Color

By clicking on this button you may select the color which will be used when drawing the axis, its tick marks, grid and axis labels.

Font

By pressing this button you open a font dialog, where font and text color can be set for the axis. Unless changed from the chart's default setting these font settings will automatically follow changes in the chart's default font, as set on the **Font** page of the **Chart Properties**.

Width

Sets the width of the axis and its tick marks.

Scale

In the **Scale** group you can set scale properties for the axis. Normally the minimum and maximum values of the axis, and the distance between tick marks, are calculated dynamically depending on the current data set. Each of these values can, however, be individually locked.

Static Min

By selecting this check box you can set a fixed minimum value for the axis. The minimum value may be entered as a calculated formula (see page 467).

Static Max

By selecting this check box you can set a fixed maximum value for the axis. The maximum value may be entered as a calculated formula (see page 467).

Static Step

By selecting this check box you can set a fixed interval for tick marks on the axis. The value may be entered as a calculated formula (see page 467). If this setting results in more than 100 tick marks, the value used will be modified.

Position

Assigns the expression to the **Left** or **Right** (**Bottom** or **Top** in horizontal mode) y-axis.

Split Axis

By selecting this check box, the y-axis will be split in two parts while still being drawn as one continuous line. This creates an impression of two charts sharing a common axis. Expressions being assigned to **Left(Bottom)** under **Position** will be drawn on the primary part of the axis (normally the upper part of the chart) while expressions being assigned to **Right(Top)** under **Position** will be drawn on the secondary part of the axis (normally the lower part of the chart).

Primary (%)

Defines the percentage of the available axis length that will be used for the primary part of the axis when **Split Axis** is used.

Dimension Axes

In the **Dimension Axes** group you select settings for the x-axes.

Primary Dimension Labels

Select one of the radio buttons to display labels of the first chart dimension labels horizontally, tilted, or vertically.

Secondary Dimension Labels

Select one of the radio buttons to display labels of the second chart dimension labels (clustered bars only) horizontally, tilted, or vertically.

Continuous

Sets the x-axis to be continuous, i.e. it will be interpreted numerically and shown with numerically correct intervals.

Forced 0

Fixes the left edge of the chart to zero on the x-axis. This option is not available when a logarithmic axis is used.

Hide Axis

Hides the x-axis.

Show Grid

Shows a grid in the chart originating from the tick marks of the x-axis. The grid lines can be formatted using the **Grid Style** and **Grid Color** settings (see below).

Show Minor Grid

Shows minor lines in-between the grid lines. Only available if **Show Grid** is selected. The grid lines can be formatted using the **Grid Style** and **Grid Color** settings (see below).

Stagger Labels

Select this option to stagger x-axis labels when they become too many to show side by side.

Reverse Stagger

X-axis labels are normally staggered bottom-up from left to right. When marking this check box, the stagger will be reversed to top-down.

Axis Color

By clicking this button you may select the color to be used when drawing the axis, its tick marks, grid and axis labels.

Font

By pressing this button you open a font dialog where font and text color can be set for the axis. Unless changed from the chart's default setting these font settings will automatically follow changes in the chart's default font as set on the **Font** page of the chart properties.

Width

Sets the width of the axis and its tick marks.

Scale

In the **Scale** group you can set scale properties for the x-axis, provided that it is set to **Continuous**. Normally the minimum and maximum values of the axis and the distance between tick marks are calculated dynamically depending on the current data set. Each of these values can however be individually locked.

Static Min

By selecting this check box you can set a fixed minimum value for the continuous x-axis. The min value may be entered as a calculated formula (see page 467).

Static Max

By selecting this check box you can set a fixed maximum value for the continuous x-axis. The max value may be entered as a calculated formula (see page 467).

Static Step

By selecting this check box you can set a fixed interval for tick marks on the continuous x-axis. The value may be entered as a calculated formula (see page 467). If this setting results in more than 100 tick marks, the value used will be modified.

Backcast

By checking this box and entering an integer in the edit box, you force the x-axis to be extended to the left of the actual data points plotted. This is useful when using trend-lines. The backcast line will be dotted. The value may be entered as a calculated formula.

Forecast

By checking this box and entering an integer in the edit box, you force the x-axis to be extended to the right of the actual data points plotted. This is useful when using trend-lines. The forecast line will be dotted. The value may be entered as a calculated formula.

Grid Style

Select a grid line style for the chart in the drop-down.

Grid Color

Click the button to select a color for grid lines in the chart.

Synchronize Zero Level for Expression-Axes

Synchronizes the zero level of the two y-axes (left/right or top/bottom).

Colors

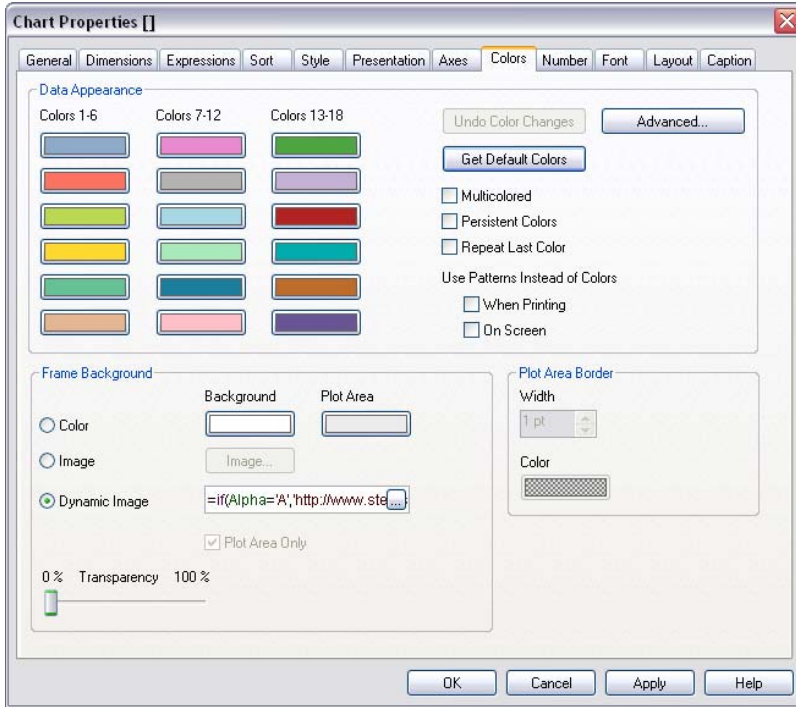


Figure 14. The Colors page in the Chart Properties dialog

In the **Colors** page, settings are made for the colors to be used when drawing the chart. The page consists of two main parts, one for the data plotting and one for the background.

Data Appearance

In the **Data Appearance** group color settings are made for the data points plotted in the chart. QlikView provides 18 different colors, which are assigned to the dimension field values according to their original load order.

Color 1-18

Here you can set the colors for each of the 18 available colors in the color map. The colors can be defined as solid col-

ors or gradients via the **Color Area** dialog (see page 23 in Book II) that opens when clicking the color button.

Undo Color Changes

Resets the color map to the settings it had upon entry to the dialog.

Get Default Colors

Resets the color map to the QlikView default settings.

Advanced...

This button opens the **Advanced Color Map** dialog (see page 73) where color maps can be set and retrieved on sheet, document, user and QlikView default level.

Multicolored

Assigns different colors to the different bars (lines, symbols or pie slices in other types of charts). If this check box is not marked, all the bars will be displayed in the same color.

Persistent Colors

Locks the colors assigned to each data point even if selections reduce the total number of data points. In this mode color representation will never change for a given data point, but you run the risk of having two adjacent bars or slices with the same color for different data points.

Repeat Last Color

Select this check box to assign the last (18th) color to field values loaded as number 18 and forward. If you leave this check box unselected, the 19th value (according to the original load order) will get the first color, the 20th value the second color, etc.

Use Patterns Instead of Colors

Mark one or two of these check boxes to use patterns instead of colors **On Screen** or **When Printing**.

Frame Background

In the **Frame Background** group color settings are made for the plot area background and the background of the area surrounding the plot area.

Color

The chart will be plotted with a colored background. Different colors can be set for the plot area and the surrounding area.

Background

The color used for the background around the plot area or, in the case of some charts, for the entire chart background. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button. The default color is white.

Plot Area

The color used for the plot area of the chart. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the button. The default color is light gray. This setting is unavailable for pie charts, block charts, funnel charts and radar charts.

Image

The chart will be plotted with an image background. Change image by pressing the button.

Dynamic Image

Enter a calculated expression to show dynamic background images that change with the selection. Available for bar, line, combo, scatter and grid charts.

Plot Area Only

When an image background is selected it can extend over the entire background or, if this check box is selected, be confined to the plot area. This option only applies when the **Image** option is chosen above.

Transparency

The chart will be plotted with a transparent background. Objects behind the chart will be visible through it.

Plot Area Border

In the **Plot Area Border** subgroup settings are made for the border rectangle normally drawn around the actual plot area. These settings are unavailable for pie charts, block charts, funnel charts and radar charts.

Width

Here you may set the line width of the rectangle drawn around the plot area. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The border can be taken away by setting the width to 0.

Border Color

Click the colored button to change the color of the rectangle. Default is black.

The Advanced Color Map dialog

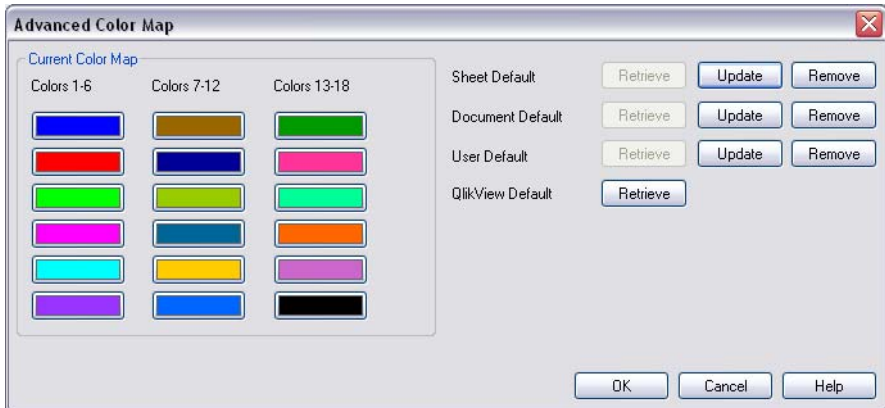


Figure 15. The Advanced Color Map dialog

In the **Advanced Color Map** dialog you may retrieve, set or clear color settings for several default levels.

Current Color Map

When the dialog is opened the color map of the current chart is shown. You may change the color map by clicking individual colors or by pressing the **Get** button for **Sheet Default**, **Document Default**, **User default** or **QlikView Default**.

Sheet Default

Sets, retrieves or clears the default color map for the current sheet. Press **Retrieve** to retrieve the current sheet default color map (option only available if a sheet default is available). Press **Update** to apply the current color map as sheet default. Press **Remove** to clear the default for the current sheet (option only available if a sheet default is available).

Document Default

Sets, retrieves or clears the default color map for the current document. Press **Retrieve** to retrieve the current document default color map (option only available if a document default is available). Press **Update** to apply the current color map as document default. Press

Remove to clear the default for the current document (option only available if a document default is available).

User Default

Sets, retrieves or clears the default color map for the current sheet. Press **Retrieve** to retrieve the current user default color map (option only available if a user default is available). Press **Update** to apply the current color map as user default. Press **Remove** to clear the default for the current user (option only available if a user default is available).

QlikView Default

Press **Retrieve** to retrieve the QlikView default color map (option only available if a sheet default is available). This default cannot be changed.

OK

Returns to the **Color** page of the **Chart Properties** dialog and applies changes in the current color map back to the color map in that dialog.

Cancel

Returns to the **Color** page of the **Chart Properties** dialog without applying changes in the current color map back to the color map in that dialog.

Help

Opens context-specific help.

Number

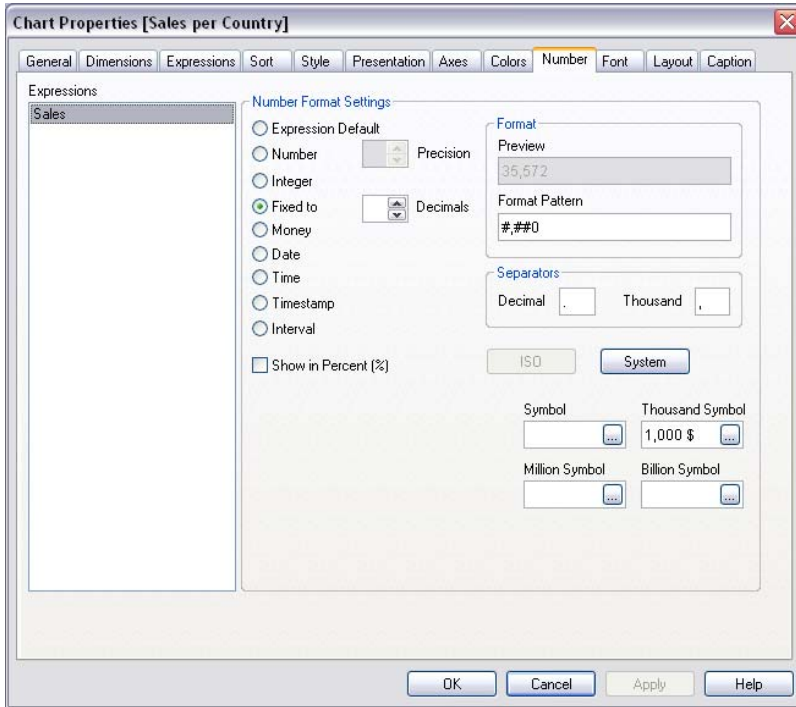


Figure 16. The Number page in the Chart Properties dialog

In the **Number** page, the display number format can be set. It is possible to set the date, time and number format for the different axes and expressions. If the x-axis is set to **Continuous** on the **Axes** page, the x-axis dimension is also treated as an expression.

Expressions

A list of the expressions in the chart. Mark the expression for which you want to set the number format. Several expressions can be marked simultaneously.

For an explanation of the other controls, see page 301 in Book II.

Font

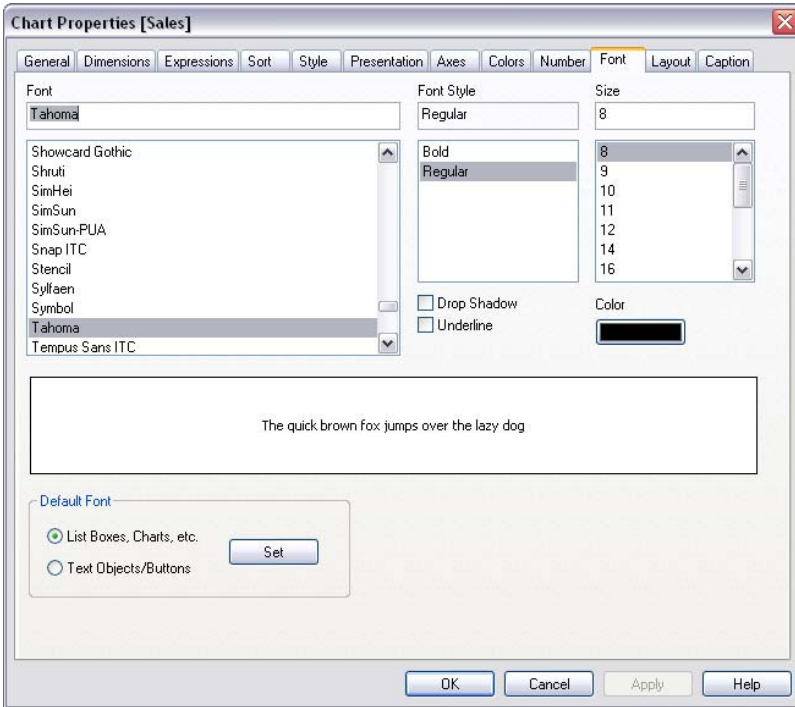


Figure 17. The Font page in the Chart Properties dialog

In the **Font** page, you can change the font of the text in the chart.

The font chosen will be applied to all text in the chart, which has not been set to a font of its own in the separate Font controls provided for e.g. title, axes or legend. You can specify the font, style and size of the text. Additional settings are:

Drop Shadow

If this check box is marked a drop shadow will be added to the text.

Underline

If this check box is marked the text will be underlined.

Color

Opens the **Color Area** dialog, see page 23 in book II.

In the **Default font** group it is possible to set the default font for different types of sheet objects.

Layout

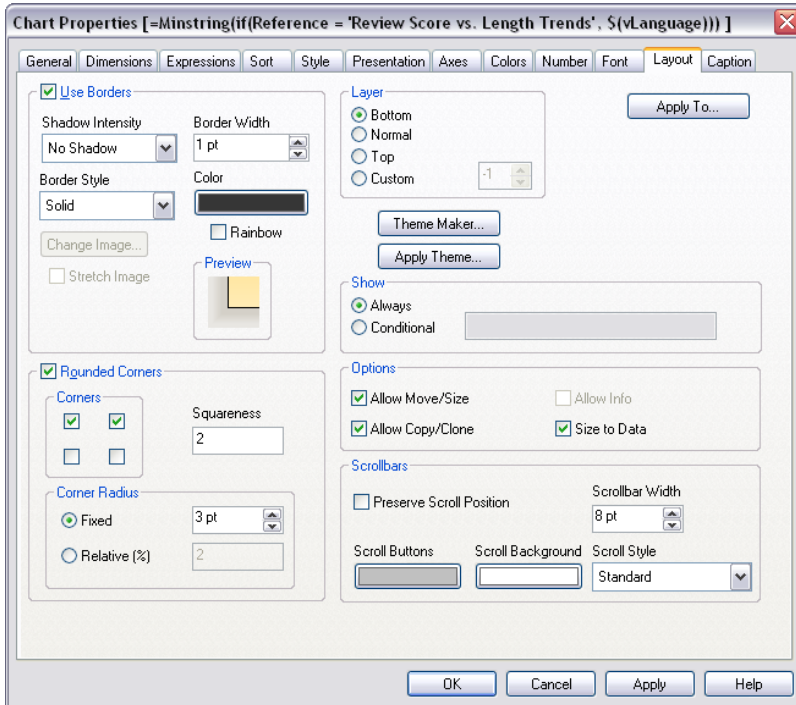


Figure 18. The Layout page in the Chart Properties dialog

In the **Layout** page you can specify how the bar chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside.

The **Layout** page is common to all sheet objects. When **Simplified** is the chosen **Styling Mode** in **Document Settings, General** page, there is no choice of border type or corner settings, there is only the **Shadow Intensity** drop-down menu and the **Border Width** setting.

Use Borders

Mark this check box in order to use a border around the sheet object. You specify the type of border by selecting in the drop-down menu. The **Shadow Intensity** drop-down menu makes it possible to set the intensity of the shadow that surrounds the sheet objects. There is also the choice of **No Shadow**.

Border Width

Here you can set the width of the border. Available for all border types. When setting the width of image borders, make sure the width corresponds to the corner width defined in the image. The width can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Border Style

There are the following predefined border types:

Solid

A solid unicolored border.

Depressed

Border giving the impression of depressing the sheet object from the background.

Raised

Border giving the impression of raising the sheet object from the background.

Walled

Border giving the impression of a wall around the sheet object.

Image

Border specified by custom image.

Change Image

If the **Image** border type is selected, you can click this button to set the image you wish to use.

Stretch Image

If the **Image** border type is used and this check box is marked the image pattern will be stretched to fit to the entire space between the corners. If it is not marked, as many copies as possible of the pattern will be displayed, tiled.

For an example of creating an image border, see page 63 in Book II.

Most of the border types can be specified further with the help of the following settings:

Color

Click this button to open a dialog in which you can choose an appropriate base color from the color palette for all border types except **Image**.

Rainbow

Creates a rainbow-colored border for all border types except **Image**. The rainbow will start with the selected base color on top of the sheet object.

Rounded Corners

Here settings can be made for the general shape of the sheet object. If the check box is marked, it is possible to use rounded shapes for the corners of the sheet object. It is possible to achieve shapes from rectangular via super elliptic to elliptic/circular.

Corners

By marking each of these four check boxes, the rounding options are turned on for the respective corners of the sheet object. Corners for which the check box remain unmarked will be drawn rectangular.

Squareness

A number between 2 and 100 defining the general shape of the rounded corners. The number 2 corresponds to a perfect ellipse (or circle for a 1:1 aspect ratio), whereas higher numbers move over super elliptic shapes towards a rectangle.

Corner Radius

This setting determines the radius of the corners in fixed distance or as a percent of the total quadrant. This setting lets you control the extent to which the corners will be affected by the underlying general shape set under **Squareness**. A fixed distance can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Layer

In the **Layer** group, the sheet object can be assigned one of four layers on the sheet.

Bottom

A sheet object with the **Bottom** layer property can never obscure sheet objects in the **Normal** and **Top** layers. It can only be placed on top of other sheet objects in the **Bottom** layer.

Normal

When created, sheet objects reside in the normal (middle) layer. A sheet object in the **Normal** layer can never be

obscured by sheet objects in the **Bottom** layer and can never obscure sheet objects in the **Top** layer.

Top

A sheet object in the **Top** layer can never be obscured by sheet objects in the **Normal** and **Bottom** layers. Only other sheet objects in the **Top** layer can be placed on top of it.

Custom

The **Top**, **Normal** and **Bottom** layers correspond to internally numbered layers 1, 0 and -1 respectively. In fact all values between -128 and 127 are accepted. Choose this option to enter a value of your choice.

Theme Maker...

Opens the **Theme Maker** dialog (see page 275 in book II) for definition of layout themes.

Apply Theme...

Opens a file browser dialog for opening a QlikView layout theme file to be applied to the sheet object. See page 275 for more information on QlikView layout themes.

Show

In the **Show** group, it is possible to specify a condition under which the sheet object is shown.

Always

The sheet object will always be shown.

Conditional

The sheet object will be shown or hidden depending on a condition expression which will be evaluated each time the sheet object is to be drawn. The sheet object will only be visible when the condition returns true.

Users with Admin privileges for the document can override all show conditions with the **Show All Sheets and Sheet Objects** in the **Security** page of **Document Properties** (see page 42). This functionality can be toggled by pressing CTRL+SHIFT+S.

Options

In the **Options** group, it is possible to disallow moving/resizing of the sheet object.


Allow Move/Size

If this check box is deselected, it will become impossible for the user to move or resize the sheet object.

Allow Copy/Clone

If this check box is deselected it will become impossible for the user to copy or clone the sheet object.

Allow Info

By default, an info icon  will appear in the top right corner of the sheet object if information is linked to the selected field value. Mark this check box if you do not wish the icon to be displayed. This option is available only for list boxes and multi boxes.

Size to Data

Normally the borders around all table sheet objects in QlikView will shrink when selections cause the size of the table to be less than allocated size for the sheet object. By deselecting this check box this automatic adjustment of size will be turned off leaving any surplus space blank.

Scrollbars

In this group you can set the appearance of scroll bars in the QlikView sheet object. When **Simplified** is the chosen **Styling Mode**, the scrollbar settings are limited.

Preserve Scroll Position

When marking this box and the corresponding box in **User Preferences, Objects** page, QlikView will try to preserve the scroll position after a selection in table objects and charts with an x-axis scroll bar.

Scroll Background

Sets the scroll bar background color. Select a color by clicking the button.

Scroll Buttons

Sets the scroll button color. Select a color by clicking the button. Note that medium gray tones often render the best results for scroll bars.

Scroll Bar Width

Sets the scroll bar width. The width can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Scroll Style

Sets the style of the scroll bar.

Apply to...

Opens the **Caption and Border Properties** dialog (see below), where you can set where to apply the properties you set on the **Layout** page.

Caption and Border Properties



Figure 19. The *Caption and Border properties page*

In this dialog you set in where in the document your caption and border properties should apply.

Apply Properties to...

Mark this check box to apply your settings to other than the current object.

Mark one of the following radio buttons.

Objects on this sheet

Applies your settings to objects on the current sheet only.

Objects in this document

Applies your settings to objects of the entire document.

Mark one of the following radio buttons:

Only this object type

Applies your settings to all objects of this type.

All object types

Applies your settings to all objects.

Set as default for new objects in this document

Mark this check box to use your settings as default for all new objects in the current document.

Caption

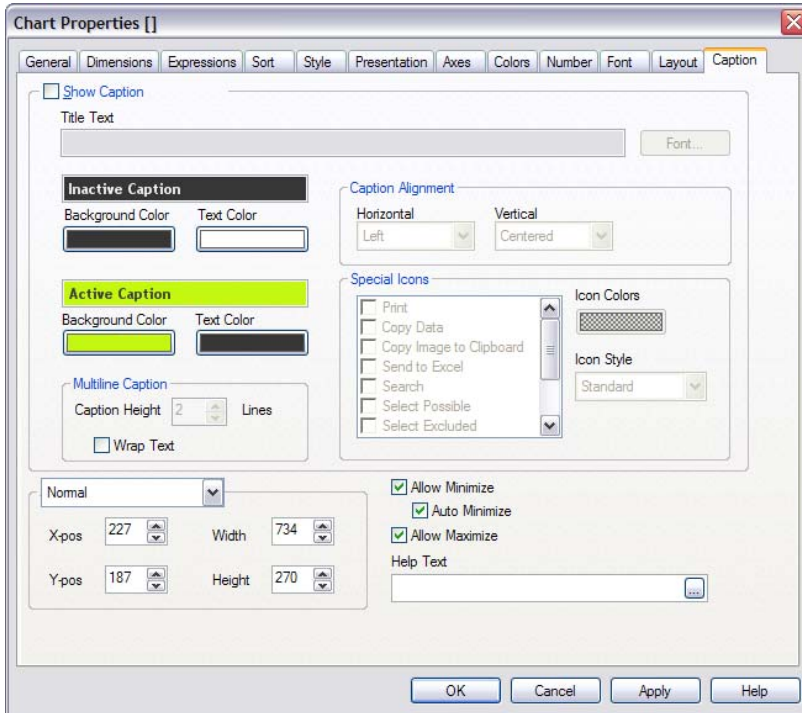


Figure 20. The Caption page in the Chart Properties dialog

In the **Caption** page advanced settings can be made for the caption. Background and foreground text color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction.

Show Caption

If this check box is marked, a caption will be drawn at the top of the sheet object. List boxes and other "box objects" will have caption turned on by default, whereas buttons, text objects and line/arrow objects will have it turned off.

Title Text

The text to be displayed in the sheet object caption. This setting is identical to the **Windows Title** setting found in the **General** page of the properties dialogs of a number of sheet objects. However, for objects such as the text objects, this is the only place where the cap-

tion text can be edited. The **Font...** button opens the **Font Dialog** where the font of the caption text can be set.

Inactive Colors / Active Colors

In this group you can set the colors of the caption in its different states. A preview pane shows the current settings of the selected state.

Background Color

Sets the background color. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the color button.

Text Color

Click the colored button to select a color.

Multiline Caption

In this group you can specify multiple line captions.

Caption Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the caption should have.

Wrap Text

If this check box is marked, the caption will be wrapped in two or more lines.

Normal/Minimized Size/Position

In this group you can set exact pixel size and position of the QlikView sheet object in its normal or minimized state. Select state in the drop-down control.

X-pos

Sets the horizontal position of the left side of the sheet object in relation to the sheet's left edge. The setting is made in pixels.

Y-pos

Sets the vertical position of the top side of the sheet object in relation to the sheet's upper edge. The setting is made in pixels.

Width

Sets the width of the QlikView sheet object in pixels.

Height

Sets the height of the QlikView sheet object in pixels.

Caption Alignment

In this group you set text alignment for caption text.

Horizontal

Sets the horizontal alignment of the caption text. The options are **Left**, **Center** and **Right**.

Vertical

Sets the vertical alignment of the caption text. The options are **Top**, **Center** and **Bottom**. This setting is only relevant when **Multiline Caption** is used.

Special Icons

Many of the object menu commands of the sheet objects can be configured as caption icons. Select commands to be shown as caption icons by marking the check box to the left of each command in the list.

Icon Colors

Sets the background color of the caption icons. The color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when clicking the color button. This setting applies to all icons in the sheet object, not only the special caption icons.

Icon Style

Sets the icon style. Select a style in the drop-down control. This setting applies to all icons in the sheet object, not only the special caption icons.

Note Use the special caption icons with care. One or two of them may provide great help for the end-user while too many may create a “Christmas tree” that severely degrades usability.

Allow Minimize

If this check box is marked, it is possible to turn the sheet object into an icon. The minimize icon will appear in the sheet object caption.


Auto Minimize

This setting, which makes it possible to toggle between sheet objects, should be applied to several sheet objects (preferably positioned on top of each other) on one and the same sheet. Only one of the sheet objects will have its full size at any given time: as soon as you restore a minimized sheet object with auto minimize set, all other sheet objects with auto minimize set will turn into icons.

Allow Maximize

If this check box is marked, it is possible to enlarge the sheet object to fill the sheet. The maximize icon will appear in the sheet object caption.

Help Text

In this field you can type an optional help message. The text will be displayed when you position the cursor on the help icon  in the top right corner of the sheet object. The text may be a calculated formula (see page 467).

3 LINE CHART

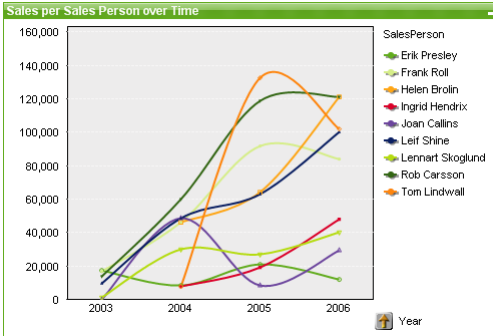


Figure 21. An example of a Line chart

Line charts present data as lines between value points, as value points only or as both lines and value points. Line charts are useful when you want to show changes or trends.

3.1 Creation

Line charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.

If you want to create a simple line chart and find the great number of different settings overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu (**QuickChart**), or click the **QuickChart** button in the toolbar. The wizard appearing will help you to create a chart in a quick and easy way. For more information, see page 265.

3.2 Line Chart Object Menu

The line chart **Object** menu is found as the **Object** menu when a line chart is active. It can also be opened as a context menu by clicking with the right mouse button on a line chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the charts static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Clear All Selections

Deselects the values of all the fields in the line chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the chart.

3.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31).

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39).

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

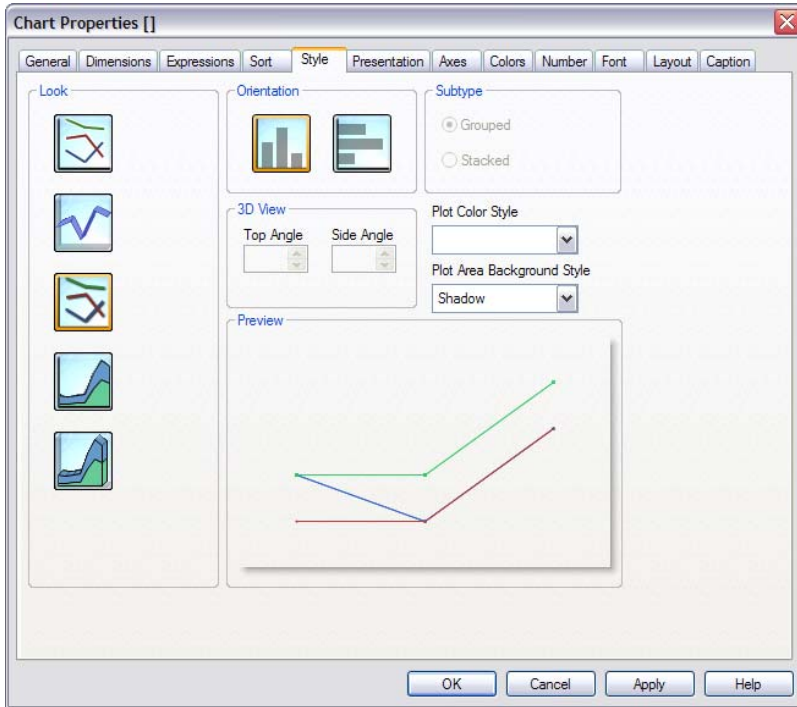


Figure 22. The Style page for line charts in the Chart Properties dialog

In this page you select the basic style for the line chart.

Look

Select one of the available styles.

Orientation

Here you can set the orientation of the chart to either vertical or horizontal as indicated by the icons.

3D View

The settings in this group define the angle from which the chart is viewed in 3D modes.

Top Angle

Defines the vertical angle of the 3D view. The value must be an integer between 0 and 30.

Side Angle

Defines the side angle of the 3D view. The value must be an integer between 0 and 45.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors page** will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all lines.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

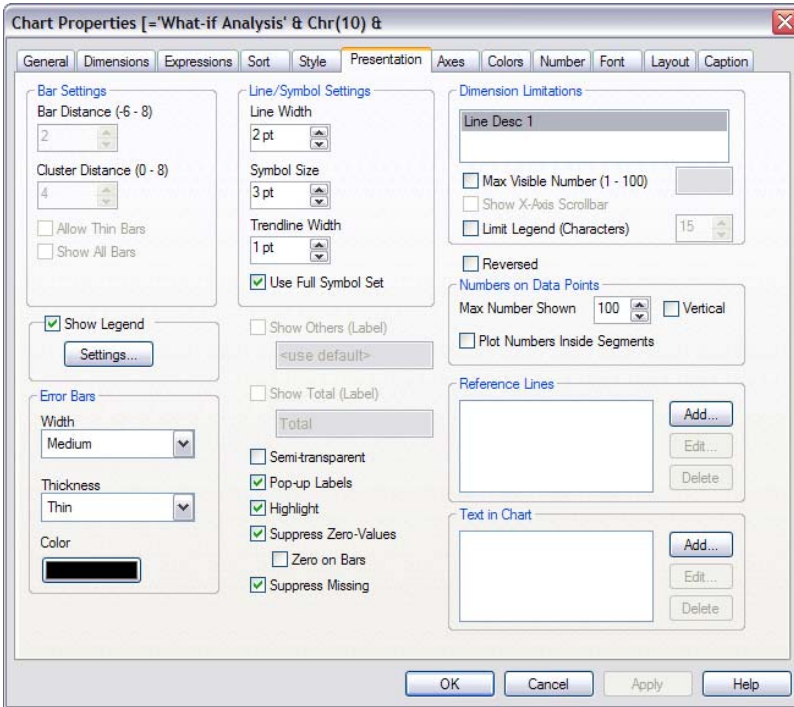


Figure 23. The Presentation page for line charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the line chart is presented on the screen.

Show Legend

Mark this check box if you want the legend to be shown in the chart when applicable. If the chart is dimensionless, but has several expressions, unchecking this box will show the expressions on the axis instead.

Settings...

Opens the **Legend Settings** dialog (see page 60), where advanced settings can be made for the chart legend.

Error Bars

In this group you can set formatting options for error bars, if such are used in the chart.

Width

Specifies the width of the error bars. You may choose between **Narrow**, **Medium** and **Wide**.

Thickness

Specifies the thickness of the error bars. You may choose between Thin, Medium and Thick.

Color

Click the button to select a color for the error bars.

Show Others (Label)

Mark this check box to display a separate x-axis value for 'Others', i.e. an expression value for all dimension values excluded by a limitation under **Dimension Limitations-Max Number**. The 'Others' value will only be meaningful for expressions where **F(x) of Rows** total is meaningful (see “**Expressions**” page 39) and the value will be calculated using the function specified under that option. A label for the 'Others' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Others** label found in the **Objects** page of the **User Preferences** will be used (see page 107 in Book I). Only symbol and no line will be plotted for 'Others'.

Show Total

Mark this check box to display a separate x-axis value for 'Total', i.e. an expression total for all dimension values. The 'Total' value will only be meaningful for expressions where **F(x) of Rows** total is meaningful (see “**Expressions**” page 39) and the value will be calculated using the function specified under that option. A label for the 'Total' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Total** label found in the **Objects** page of the **User Preferences** will be used (see page 107 in Book I). Only symbol and no line will be plotted for 'Total'.

Semi-transparent

By selecting this check box, the line chart will be drawn semi-transparent when using area style.

Pop-up Labels

By selecting this check box the values of the dimension(s) and expression(s) will appear as a pop-up balloon when hovering with the cursor over data point in the chart.

Highlight

If this check box is marked hovering with the mouse over a line or symbol in it will result in the line or symbol being highlighted for better overview. The highlighting applies also in the legend where applicable.

Suppress Zero-Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Zero on Bars

This setting is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Point** is selected for the chart expression under **Display Options** on the **Expressions** page, zero values will appear as text above the data points. In other cases zero values will be suppressed.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Max Number

Select this check box to limit the number of dimension values to be shown at any one time. Enter the maximum number of values in the scroll box.

Show X-Axis Scroll Bar

Select this check box to show a scroll control in the place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values to be shown. The number of values shown at any one time will be the number set under **Max Number**. This option is only available when **Max Number** is selected.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Reversed

When this check box is marked, the values shown in the chart will be reversed, i.e. if the 10 highest values are shown and the box is checked the chart will then show the 10 lowest values.

Values on Data Points

In the **Values on Data Points** group you can set display options for values on data points, provided that this option has been selected for one or more chart expressions under **Plot Options** in the **Expression** page of the chart's properties.

Vertical

Shows the values vertically.

Max value Shown

In this box you can specify an upper limit for how many data points to show values for in the chart. If no limit is specified, values will be shown for all data points, which may affect the readability of the chart.

Plot Values Inside Segments

Marking this check box will plot values on data points inside the segments instead of on top of them.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the chart plot area from a given point on a continuous x- or y-axis. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 139) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click this button in order to edit its properties in the **Reference Lines** dialog (see page 139). Double-clicking on the reference line in the list has the same effect.

Delete

Mark an existing reference line in the list and click this button in order to delete it from the list.

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog (see page 64) where you may create and edit free-floating texts to be displayed in the chart.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog (see page 64). Double-clicking the text in the list has the same effect.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Line/Symbol Settings

In the **Line/Symbol Settings** group you can set display options for lines and symbols in the chart.

Line Width

Sets the line width for lines. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). This option is available only when lines are used.

Symbol Size

Sets the size of the symbols, when such are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Trendline Width

Here you can set the width of the trendlines by entering an appropriate number. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Use Full Symbol Set

If this check box is marked, different kinds of symbols available will be used to distinguish data points from each other. If it is not marked, all the symbols will be drawn as plus signs.

Axes

In the **Axes** page, the display properties of the x- and y-axes settings are set. This page is identical with the **Axes** page of the bar chart (Figure 13 on page 65).

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the line chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).

4 COMBO CHART

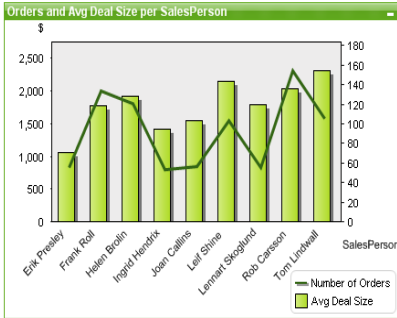


Figure 24. An example of a Combo chart

The combo chart allows you to combine the features of the bar chart with those of the line chart: you can show the values of one expression as bars while displaying those of a second expression as lines or symbols.

4.1 Creation

Combo charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.

4.2 Combo Chart Object Menu

The combo chart **Object** menu is found as the **Object** menu when a combo chart is active. It can also be opened as a context menu by clicking with the right mouse button on a combo chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the charts static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Set Reference

By choosing this option you set a chart reference, i.e. a fixed plot of the chart with the current selections. When further selections are made in the document the reference plot remains, dimmed in the background. Chart axes etc. will be adjusted to always include the maximum of the background data set and the current data set. The current data set is always plotted on top of the reference plot, i.e. some portions of the reference plot may be obscured by the current data set plot. The way the background is dimmed can be controlled via the **Reference Mode** setting on the **General** page of the **Chart Properties** dialog. The display of reference chart plots is only possible in some chart types, e.g. bar charts, line charts, combo charts, radar charts, scatter charts, grid charts and gauge charts with needles. It is not possible to set reference for a chart that contains a drill-down or a cyclic group. The reference will be lost when closing the document or when reloading data.

Clear Reference

This menu item replaces the **Set Reference** menu item when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Deselects the values of all the fields in the chart.

Print...

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.


Help

Opens context-specific help.

Remove

Removes the chart.

4.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc. 

Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contains a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31).

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39).

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

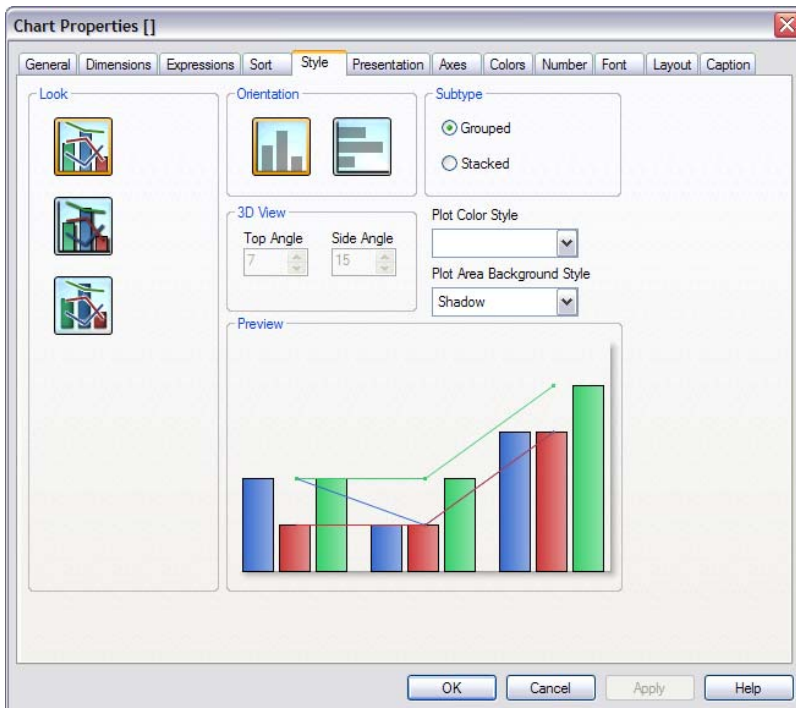


Figure 25. The Style page for combo charts in the Chart Properties dialog

In this page you select the basic style of the combo chart.

Look

Select one of the available styles.

Orientation

Here you can set the orientation of the chart to either vertical or horizontal as indicated by the icons.

Chart Subtype

In this group it is possible to choose between **Grouped** and **Stacked** mode, provided that the chart has exactly two dimensions or exactly two dimensions but more than one expression. Negative values in stacked combos are stacked separately downwards below the x-axis.

3D View

The settings in this group define the angle from which the chart is viewed in 3D modes.

Top Angle

Defines the vertical angle of the 3D view. The value must be an integer between 0 and 30.

Side Angle

Defines the side angle of the 3D view. The value must be an integer between 0 and 45.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors page** will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all bars and lines.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

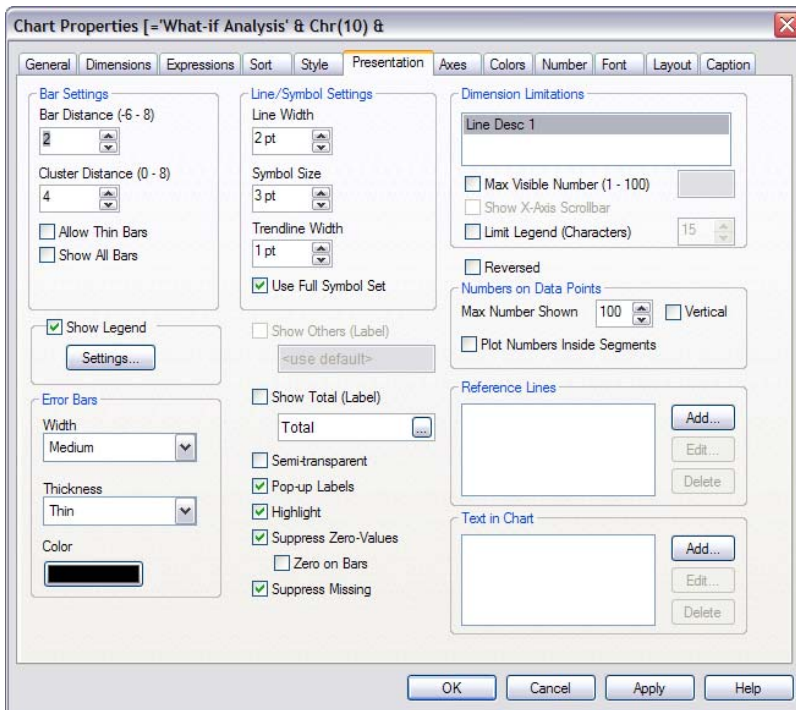
Presentation

Figure 26. The Presentation page for combo charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the combo chart is presented on the screen.

Bar Settings

In the **Bar Settings** group you can set display options for bars in the chart.

Bar Distance

By entering an appropriate number, it is possible to set the distance between the bars in the cluster. It is also possible to enter negative numbers. Values between -6 and 8 are allowed.

Cluster Distance

By entering an appropriate number, it is possible to set the distance between clusters of bars in the chart. Values between 0 and 8 are allowed.

Allow Thin Bars

For charts with non-continuous x-axis, QlikView will only display as many data points as can be accommodated in the plot area available. Remaining data points are truncated from the chart. By default bars will be drawn with a minimum width of four pixels, so that they are clearly distinguishable. Mark this check box to allow bar width down to one pixel.

Show All Bars

For charts with non-continuous x-axis, QlikView will only display as many data points as can be accommodated in the plot area available. Remaining data points are truncated from the chart. Mark this check box to force plotting of all bars. The bars will be compressed and some data points may become obscured by others.

Show Others (Label)

Mark this check box to display a separate x-axis value for 'Others', i.e. an expression value for all dimension values excluded by a limitation under **Dimension Limitations - Max Number**. The 'Others' value will only be meaningful where the **F(x) of Rows** total (see **Expression** page on page 39) and the value will be calculated using the function specified under that option. A label for the 'Others' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Others** label found in the **Objects** page of the **User Preferences** will be used (see page 107 in Book I).

Show Total

Mark this check box to display a separate x-axis value for 'Total', i.e. an expression total for all dimension values. The 'Total' value will only be meaningful for expressions where the **F(x) of Rows** total is meaningful (see **Expression** page on page 39) and the value will be calculated using the function specified under that option. A label for

the 'Total' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default Total label found in the **Objects** page of the **User Preferences** will be used (see page 107 in Book I).

Semi-transparent

By selecting this check box, the area to be filled will be drawn semi-transparent.

Line/Symbol Settings

In the **Line/Symbol Settings** group you can set display options for lines and symbols in the chart.

Line Width

Sets the line width for lines. This option is available only when lines are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Symbol Size

Sets the size of the symbols, when such are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Trendline Width

Here you can set the width of the trendlines by entering an appropriate number. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Use Full Symbol Set

If this check box is marked, different kinds of symbols available will be used to distinguish data points from each other. If it is not marked, all the symbols will be drawn as plus signs.

Pop-up Labels

By marking this check box the values of the dimension(s) and expression(s) will appear as a pop-up balloon when hovering with the cursor over data point in the chart.

Highlight

If this check box is marked hovering with the mouse over a line or symbol in it will result in the line or symbol being highlighted for better overview. The highlighting applies also in the legend where applicable.

Suppress Zero-Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Zero on Bars

This setting is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Point** is selected for the chart expression under **Display Options** on the **Expressions** page, zero values will appear as text above the data points. In other cases zero values will be suppressed.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Show Legend

Mark this check box if you want the legend to be shown in the chart when applicable. If the chart is dimensionless, but has several expressions, unchecking this box will show the expressions on the axis instead.

Settings...

Opens the **Legend Settings** dialog (see page 60), where advanced settings can be made for the chart legend.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Max Number

Select this check box to limit the number of dimension values to be shown at any one time. Enter the maximum number of values in the scroll box.

Show X-Axis Scroll Bar

Select this check box to show a scroll control in the place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values to be shown. The number of values shown at any one time will be the number set under **Max**

Number. This option is only available when **Max Number** is selected.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Reversed

When this check box is marked, the values shown in the chart will be reversed, i.e. if the 10 highest values are shown and the check box is marked the chart will then show the 10 lowest values.

Numbers on Data Points

In the **Values on Data Points** group you can set display options for values on data points, provided that this option has been selected for one or more chart expressions under **Plot Options** in the **Expression** page of the chart's properties.

Vertical

Shows the values vertically.

Max Value Shown

In this box you can specify an upper limit for how many data points to show values for in the chart. If no limit is specified, values will be shown for all data points, which may affect the readability of the chart.

Plot Values Inside Segments

Marking this check box will plot values on data points inside the segments instead of on top of them.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 60) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click this button in order to edit its properties in the **Reference Lines** dialog (see page 60). Double-clicking the reference line in the list has the same effect.

Delete

Mark an existing reference line in the list and click this button in order to delete it from the list.

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog (see page 64) where you may create and edit free-floating texts to be displayed in the chart.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog (see page 64). Double-clicking the text in the list has the same effect.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Axes

In the **Axes** page, the display properties of the x- and y-axes settings are set. This page is identical with the **Axes** page of the bar chart (Figure 13 on page 65).

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the combo chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).



5 RADAR CHART

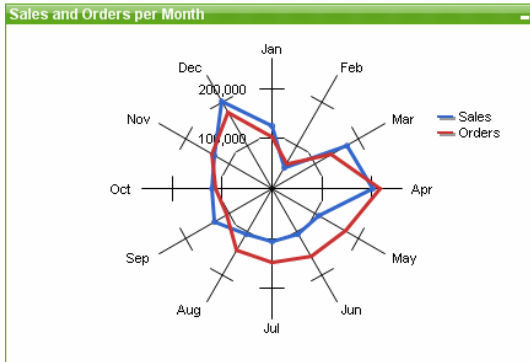


Figure 27. An example of a radar chart

Radar charts could be described as line charts where the x-axis is wrapped around 360 degrees and with one y-axis for each x-value. The result is similar to a spider web or a radar screen.

5.1 Creation

Radar charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.



5.2 Radar Chart Object Menu

The radar chart **Object** menu is found as the **Object** menu when a radar chart is active. It can also be opened as a context menu by clicking with the right mouse button on a radar chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the charts static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Set Reference

By choosing this option you set a chart reference, i.e. a fixed plot of the chart with the current selections. When further selections are made in the document the reference plot remains, dimmed in the background. Chart axes etc. will be adjusted to always include the maximum of the background data set and the current data set. The current data set is always plotted on top of the reference plot, i.e. some portions of the reference plot may be obscured by the current data set plot. The way the background is dimmed can be controlled via the **Reference Mode** setting on the **General** page of the **Chart Properties** dialog. The display of reference chart plots is only possible in some chart types, e.g. bar charts, line charts, combo charts, radar charts, scatter charts, grid charts and gauge charts with needles. It is not possible to set reference for a chart that contains a drill-down or a cyclic group. The reference will be lost when closing the document or when reloading data.

Clear Reference

This menu item replaces the **Set Reference** menu item when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.

Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.

Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...


Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Clear All Selections


Deselects the values of all the fields in the radar chart.

Copy to Clipboard


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the chart.

5.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog opens. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.

Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** buttons. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31).

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39).

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

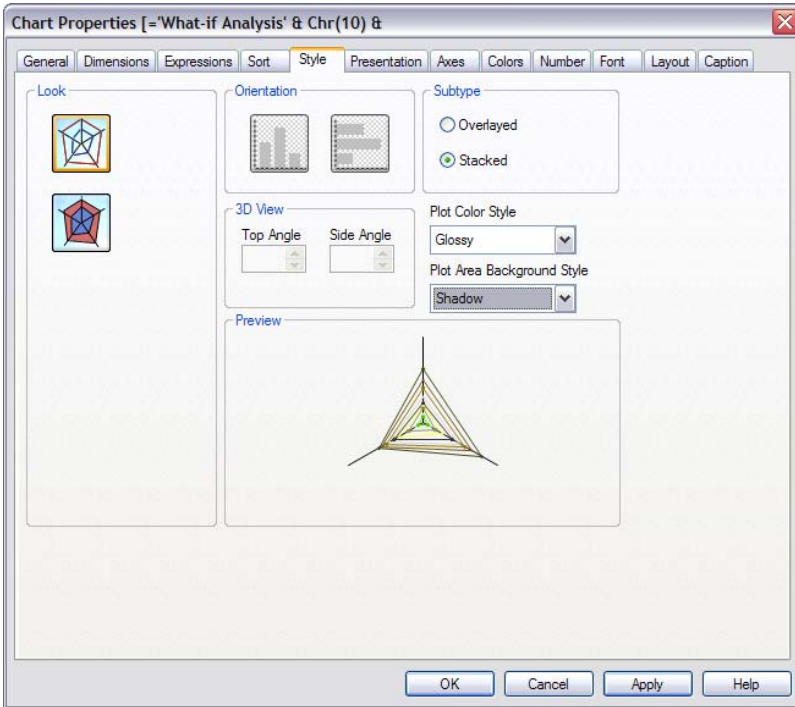


Figure 28. The Style page for radar charts in the Chart Properties dialog

In this page you select the basic style of the radar chart. The options of the **Orientation** and **3D View** groups are not available for radar charts.

Look

Select one of the available styles.

Subtype

In this group it is possible to choose between **Overlaid** and **Stacked** mode, provided that the chart has more than one dimension or more than one expression. In overlaid mode each polygon is plotted from the center of the chart. In stacked mode the polygons are plotted outside each other.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors page** will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot**

Color Style setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all segments.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

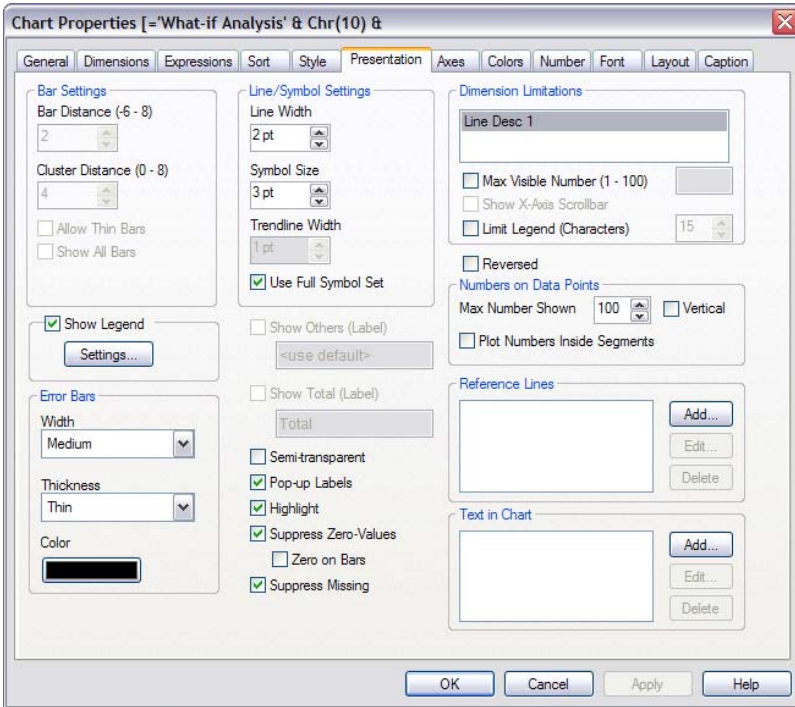


Figure 29. The Presentation page for radar charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the radar chart is presented on the screen.

Line/Symbol Settings

In the **Line/Symbol Settings** group you can set display options for lines and symbols in the chart.

Line Width

Sets the line width for lines. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The line can be taken away by setting the width to 0. This option is available only when lines are used.

Symbol Size

Sets the size of the symbols, when such are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl,

pixel), points (pt, pts, point) or docunits (du, docunit). The symbol can be taken away by setting the size to 0.

Trendline Width

Here you can set the width of the trend lines by entering an appropriate number. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The trendline can be taken away by setting the width to 0.

Use Full Symbol Set

If this check box is marked, different kinds of symbols available will be used to distinguish data points from each other. If it is not marked, all the symbols will be drawn as plus signs.

Show Legend

Select this check box if you want legend to be shown in the chart when applicable.

Settings...

Opens the **Legend Settings** dialog (see page 60 for details), where advanced settings can be made for the chart legend.

Show Others (Label)

Mark this check box to display a separate x-axis value for 'Others', i.e. an expression value for all dimension values excluded by a limitation under **Dimension Limitations - Max Number**. The 'Others' value will only be meaningful where the **F(x) of Rows** total (see **Expression** page on page 39) and the value will be calculated using the function specified under that option. A label for the 'Others' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Others** label found in the **Objects** page of the **User Preferences** will be used (see page 107 in Book I).

Show Total

Mark this check box to display a separate x-axis value for 'Total', i.e. an expression total for all dimension values. The 'Total' value will only be meaningful for expressions where the **F(x) of Rows** total is meaningful (see **Expression** page on page 39) and the value will be calculated using the function specified under that option. A label for the 'Total' data point may be typed in the edit box. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default **Total** label found in the **Objects**

page of the **User Preferences** will be used (see page 107 in Book I).

Semi-transparent

By selecting this check box, the area to be filled will be drawn semi-transparent.

Pop-up Labels

By selecting this check box the values of the dimension(s) and expression(s) will appear as a pop-up balloon when hovering with the cursor over data points in the chart.

Highlight

If this check box is marked hovering with the mouse over a line or symbol in it will result in the line or symbol being highlighted for better overview. The highlighting applies also in the legend where applicable.

Suppress Zero-Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Zero on Bars

This setting is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Point** is selected for the chart expression under **Display Options** on the **Expressions** page, zero values will appear as text above the data points. In other cases zero values will be suppressed.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Max Number

Select this check box to limit the number of dimension values to be shown at any one time. Enter the maximum number of values in the scroll box.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Reversed

When this check box is marked, the values shown in the chart will be reversed, i.e. if the 10 highest values are shown and the check box is marked the chart will then show the 10 lowest values.

Values on Data Points

In the **Values on Data Points** group you can set display options for values on data points, provided that this option has been selected for one or more chart expressions under **Plot Options** in the **Expression** page of the chart's properties.

Vertical

Shows the values vertically.

Max Value Shown

In this box you can specify an upper limit for how many data points to show values for in the chart. If no limit is specified, values will be shown for all data points, which may affect the readability of the chart.

Plot Values Inside Segments

Marking this check box will plot values on data points inside the segments instead of on top of them.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the chart plot area for a given point on the y-axes. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 62 for details) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click this button in order to edit its properties in the **Reference Lines** dialog (see page 62 for details). Double-clicking the reference line in the list has the same effect.

Delete

Mark an existing reference line in the list and click this button in order to delete it from the list.

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog where you may create and edit free-floating texts to be displayed in the chart. See page 64 for details about the **Text in Chart** dialog.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog. Double-clicking the text in the list has the same effect. See page 64 for details about the **Text in Chart** dialog.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Axes

In the **Axes** page, the display properties of the x- and y-axes settings are made. This page is identical with the **Axes** page of the bar chart (Figure 13 on page 65).

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. The page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the radar chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77)

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for

active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).



6 SCATTER CHART

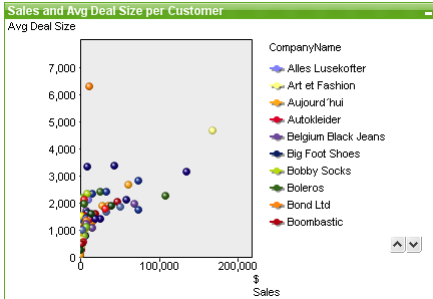


Figure 30. An example of a Scatter chart

The scatter chart presents pairs of values from two expressions. This is useful when you want to show data where each instance has two numbers, e.g. country (population and population growth).

6.1 Creation

Scatter charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.

6.2 Scatter Chart Object Menu

The scatter chart **Object** menu is found as the **Object** menu when a scatter chart is active. It can also be opened as a context menu by clicking with the right mouse button on a scatter chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the charts static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Deselects the values of all the fields in the scatter chart.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the chart.

6.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc. The scatter chart is somewhat different from



the other chart types in that it actually displays expressions on both axes while iterating over one or more dimensions.

Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contains a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions over which the chart should be iterated can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). In most cases, scatter charts should have only one or two dimensions, although more can be specified. If two dimensions are used the chart will have one data point for each combination of values in the first and second dimension.

Expressions

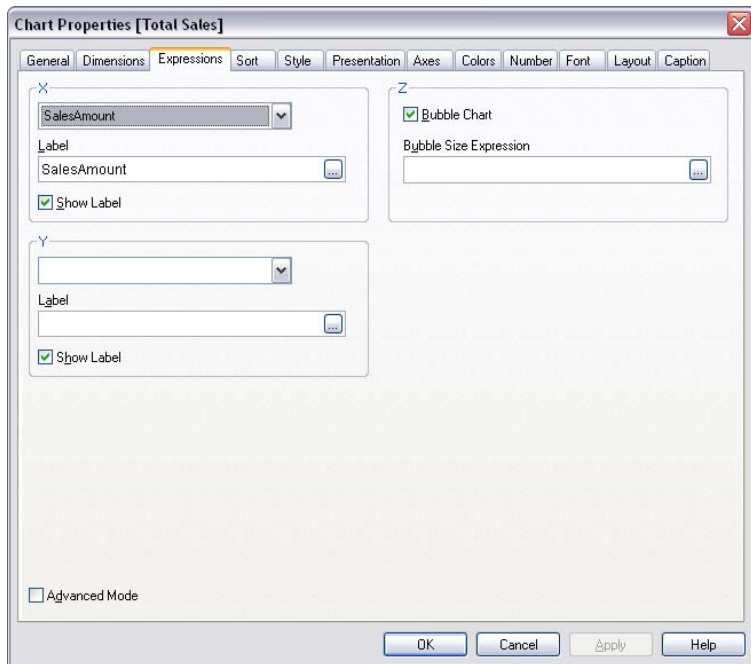


Figure 31. The Expressions page in the Scatter Chart Properties dialog

In the typical scatter chart, the x -axis and the y-axis both display an expression applying the **avg** or **only** function on single fields. The **avg** function is most likely to produce optimal results.

To make the creation of standard scatter charts easy, a simplified expression page is used. You only have to select the fields you wish to display on the axes, and possibly specify appropriate labels.

X Y

In the **X** and **Y** groups you specify the fields to be shown on the respective axes by selecting them from the respective drop-down lists.

Label

By default, the field names are set as expression labels, but you can edit them in these boxes. The label may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Show Label

Check these boxes if you want the expression labels to be displayed in the chart.

Z

In the **Z** group you specify an optional expression which will be used to calculate the size of each data point plotted in the scatter.

Bubble Chart

If this check box is selected a third (**Z**) chart expression will be used to calculate the relative size of the scatter points.

The value of the expression will determine the area of the bubble plotted.

Bubble Size Expression

This is where you enter the expression that determines the relative size of the bubble drawn at each scatter point.

Advanced Mode

Changes the **Expressions** page to the standard dialog page used for all other chart types.

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

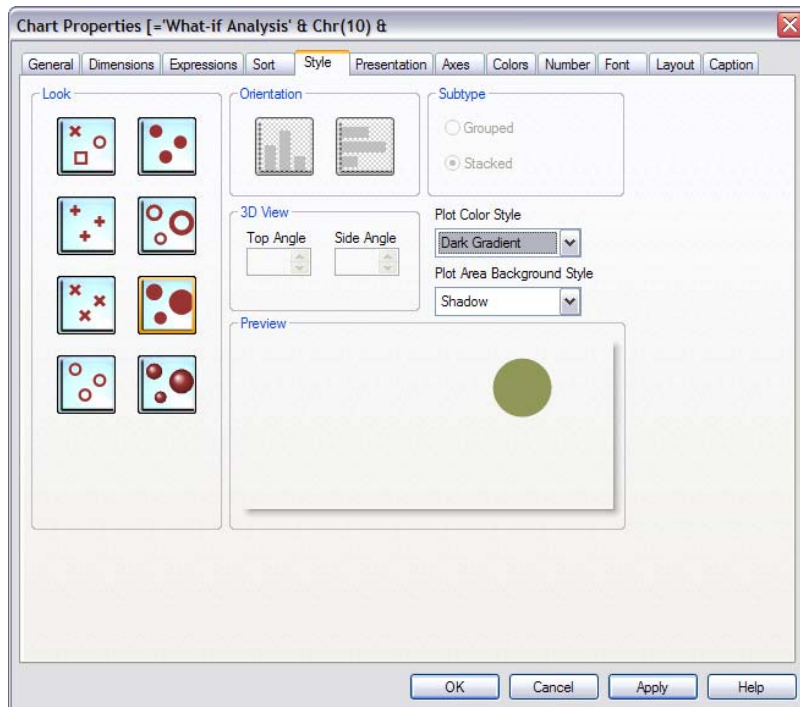


Figure 32. The Style page for scatter charts in the Chart Properties dialog

In this page you select the basic style of the scatter chart. The options of the **Orientation**, **3D View** and **Subtype** groups are not available for scatter charts.

Look

Select one of the available styles. If a bubble variant is selected a third (Z) chart expression will be used to calculate the relative size of the scatter points. The value of the expression will determine the area of the bubble plotted.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors** page will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all bubbles.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

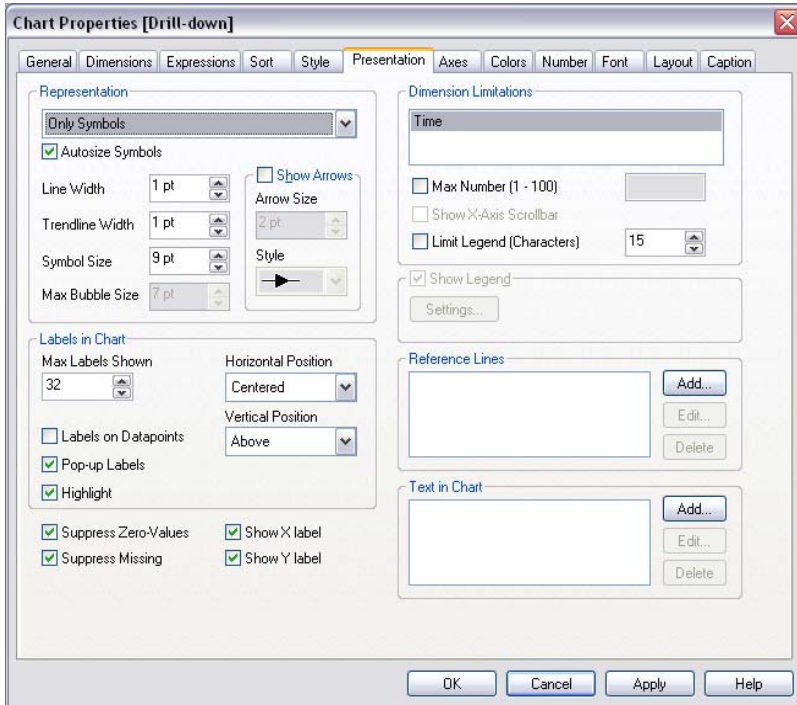


Figure 33. The Presentation page for scatter charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the scatter chart is presented on the screen.

Representation

In the **Representation** group you can set display options for scatter symbols and connecting lines in the chart. When a scatter chart has two dimensions instead of one defined on the **Dimensions** page, each value in the first dimension will result in one scatter data point for each of its associated values in the second dimension. These may then be connected by lines. In the drop-down box you can choose between drawing **Only Symbols**, **Only Lines** or **Both Lines and Symbols**.

Line Width

Sets the line width for lines, when such are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Trendline Width

Here you can set the width of the trendlines, when such are used by entering an appropriate number. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Trendlines are selected on the **Expressions** page in **Advanced Mode**.

Symbol Size

Sets the size of the symbols, when such are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Max Bubble Size

Sets the size of the largest bubble in the chart when bubble style is used. All smaller values from the Z expression will be plotted smaller in a linear relation. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The bubbles can be taken away by setting the width to 0.

Show Arrows

If this check box is marked and the scatter chart has more than one dimension defined on the **Dimensions** page arrows will be drawn on the connecting lines. The arrows will be directed between scatter points in the order defined by the sort order of the second dimension.

Arrow Size

Sets the size of the arrows, when such are used.

Style

Sets the style of the arrows, when such are used.

Suppress Zero Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Show X Label

Shows field label along the x-axis.

Show Y Label

Shows field label along the y-axis.

Labels in Chart

In the **Labels in Chart** group you can set options for showing data point labels in the chart plotting area. The labels plotted are the same as those in the legend.

Max Labels Shown

Sets a limit for how many labels that are plotted in the chart. Setting this number too high may degrade chart readability.

Labels on Data Points

Shows labels next to the scatter points inside the chart plot area.

Pop-up Labels

By selecting this check box the value(s) of the dimension(s) will appear as a pop-up balloon when hovering with the cursor over data point in the chart. This setting works independently from the **Labels on Data Points** option. The settings below in this group have no effect on pop-up labels.

Highlight

If this check box is marked hovering with the mouse over a line or symbol in it will result in the line or symbol being highlighted for better overview. The highlighting applies also in the legend where applicable.

Horizontal Position

In the drop-down box you can select whether the labels should be plotted to the left of the scatter point, to the right of the scatter point or be horizontally centered on the scatter point.

Vertical Position

In the drop-down box you can select whether the labels should be plotted on top of the scatter point, below the scatter point or be vertically centered on the scatter point.

Dimension Limitations

In this group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list, then check **Max Number** and enter an appropriate number of values to be shown.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Show Legend

Select this check box if you want legend to be shown in the chart when applicable.

Settings...

Opens the **Legend Settings** dialog (see page 60), where advanced settings can be made for the chart legend.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the chart plot area from a given point on the x-axis, the y-axis or both. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 139 for details) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click on this button in order to edit its properties in the **Reference Lines** dialog (see page 139 for details). Double-clicking on the reference line in the list has the same effect.

Delete

Mark an existing reference line in the list and click on this button in order to delete it from the list.

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog (see page 64 for details) where you may create and edit free-floating texts to be displayed in the chart.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog (see page 64 for details). Double-clicking the text in the list has the same effect.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

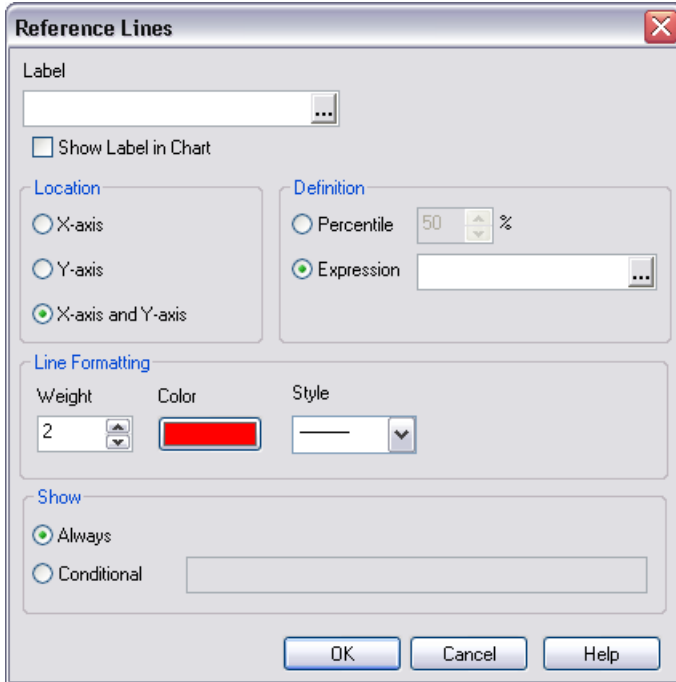
Reference Lines Dialog

Figure 34. The Reference Lines dialog of the Scatter chart

A reference line in the scatter chart is a line intersecting the chart plot area from a given point on one or both axes. It may be used e.g. to indicate a certain level, or percentiles of chart data. The reference line is only drawn if it falls within the current range of the axis from which it origins.

Legend

Here you may specify a label to be drawn next to the reference line. As default the value of **Expression** will be used.

Show Legend

Select this check box if you want the label to appear next to the reference line.

Axis

In the **Axis** group you define from which axis the reference line should origin.

X

Select this option if you want the reference line to origin from the x-axis.

Y

Select this option if you want the reference line to origin from the y-axis.

X and Y

Select this option if you want the reference line to origin from both the x- and y-axes.

Definition

In the **Definition** group the value at which the reference line should be drawn is specified. For a scatter chart it could be either at a fixed percentile of the current chart data or an arbitrary numeric expression.

Percentile

Select this option if you want the reference line to be drawn at a given percentile of the current chart data. The percentile (a value between 1 and 100) must be entered in the edit box.

Expression

Select this option if you want to specify a fixed value or an arbitrary numeric expression for where the reference line is to be drawn. The value in the edit box may be a calculated expression (see page 467).

Line Formatting

In the **Line Formatting** group you define how the reference line will look.

Weight

Specifies the width of the reference line. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The line can be taken away by setting the width to 0.

Color

By clicking on the button you may select a color for the reference line.

Style

Specifies the style of the reference line (continuous, dashed or dotted).

Show

In this group, it is possible to specify a condition under which the reference line is shown.

Always

The reference line will always be shown.

Conditional

The reference line will be shown or hidden depending on a condition expression which will be evaluated each time the chart is to be drawn. The reference line will only be visible when the condition returns true.

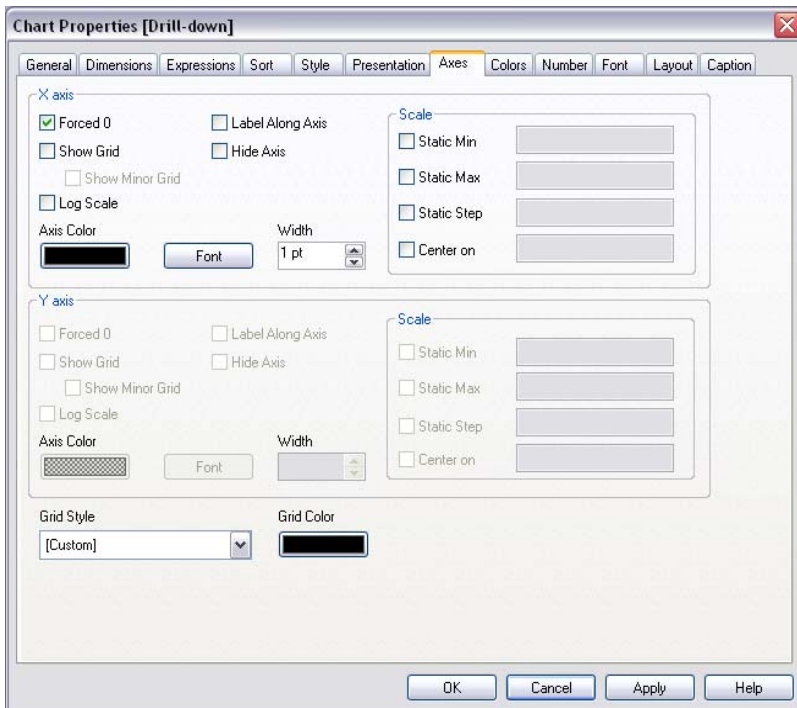
Axes

Figure 35. The Axes page for scatter charts in the Chart Properties dialog

In the **Axes** page settings for the x- and y-axes can be made.

X-axis

In the **X-axis** group you select settings for the scatter chart x-axis (horizontal axis).

Forced 0

Fixes the lower edge of the chart to zero on the axis. This option is not available when a logarithmic axis is used.

Show Grid

Shows a grid in the chart originating from the tick marks of the y-axis. The grid lines can be formatted using the **Grid Style** and **Grid Color** settings (see below).

Show Minor Grid

Shows minor lines in-between the grid lines. Only available if the **Grid Style** and **Grid Color** settings has been selected (see below).

Log Scale

Changes the y-axis to a logarithmic scale. A logarithmic scale can only be used when all data points in the charts are positive (>0).

Hide Axis

Hides the y-axis.

Label Along Axis

If this check box is marked the expression label for the axis will be drawn alongside and centered on the axis. If this option is deselected the label will be drawn to the right.

Axis Color

By clicking on this button you may select the color which will be used when drawing the axis, its tick marks, grid and axis labels.

Font

By pressing this button you will come to a font dialog where font and text color can be set for the axis. Unless changed from the chart's default setting these font setting will automatically follow changes in the chart's default font as set on the **Font** page of the chart properties.

Width

Sets the width of the axis and its tick marks.

Scale

In the **Scale** group you can set scale properties for the axis. Normally the minimum and maximum values of the axis and the dis-

tance between tick marks are calculated dynamically depending on the current data set. Each of these values can however be individually locked.

Static Min

By selecting this check box you can set a fixed minimum value for the axis. The min value may be entered as a calculated formula (see page 467). If the **Center on** value conflicts with **Static Min** and **Static Max** the **Center on** value will have priority.

Static Max

By selecting this check box you can set a fixed maximum value for the axis. The max value may be entered as a calculated formula (see page 467). If the **Center on** value conflicts with **Static Min** and **Static Max** the **Center on** value will have priority.

Static Step

By selecting this check box you can set a fixed interval for tick marks on the axis. The value may be entered as a calculated formula (see page 467). If this setting results in more than 100 tick marks, the value used will be modified.

Center on

By selecting this check box you can set a fixed value on the axis which will be positioned in the center of the plot area. The value may be entered as a calculated formula (see page 467). If the **Center on** value conflicts with **Static Min** and **Static Max** the **Center on** value will have priority.

Y-axis

In the **Y-axis** group you select settings for the scatter chart y-axis (vertical axis). The individual settings are exactly the same as for the x-axis (see above) except for the:

Label along Axis

If this check box is marked the expression label for the axis will be drawn vertically alongside and centered on the y-axis. The text will be tilted 90 degrees. If this option is deselected the expression label will be printed horizontally at the top of the axis.

Grid Style

Select a grid line style for the chart in the drop-down.

Grid Color

Click the button to select a color for grid lines in the chart.

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the scatter chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).

7 GRID CHART

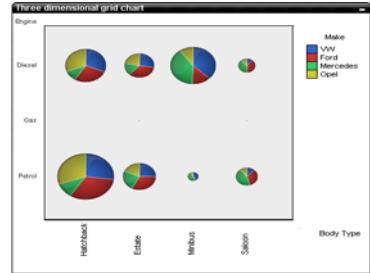
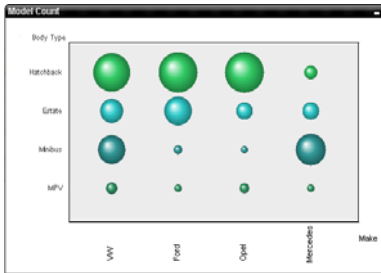


Figure 36. Two examples of a grid chart

The grid chart is similar to the scatter chart but plots dimension values on the axes and uses an expression to determine the plot symbol. A special mode makes it possible to show a third dimension in the form of small pie charts as plot symbols.

7.1 Creation

Grid charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.

7.2 Grid Chart Object Menu

The grid chart **Object** menu is found as the **Object** menu when a grid chart is active. It can also be opened as a context menu by clicking with the right mouse button on a grid chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the charts static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Deselects the values of all the fields in the grid chart.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from QlikTech's homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context-specific help.

Remove

Removes the chart.

7.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). A grid chart typically has two or three dimensions. The first dimension will be plotted on the X-axis and the second dimension on the Y-axis. If a third dimension is present it can be used to create small pie charts for each position in the grid. Further dimensions are ignored.

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39). In a grid chart only the first dimension is used when plotting the chart.

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

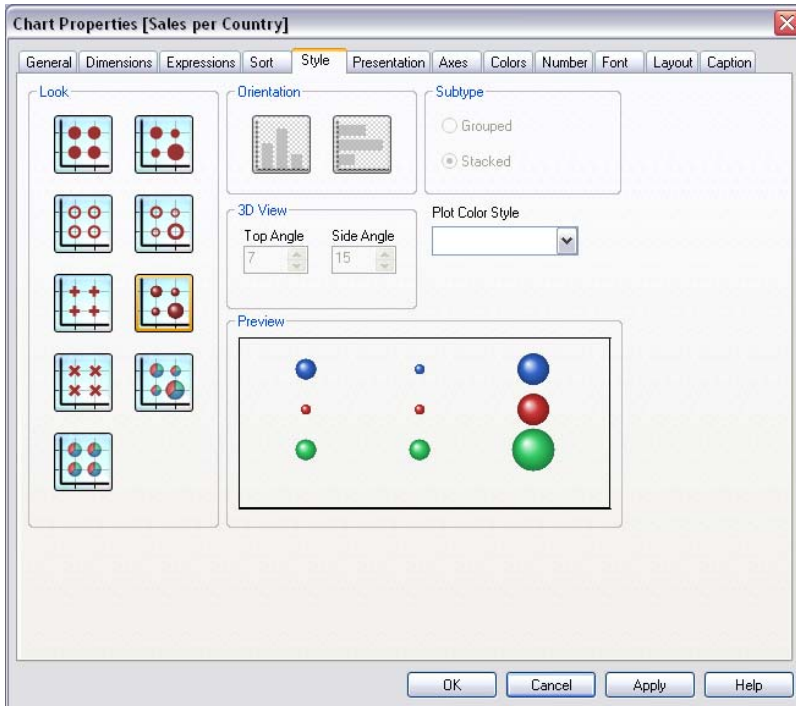


Figure 37. The Style page for grid charts in the Chart Properties dialog

In this page you select the basic style of the grid chart. The options of the **Orientation**, **3D View** and **Subtype** groups are not available for grid charts.

Look

Select one of the available styles.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors** page will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all segments.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

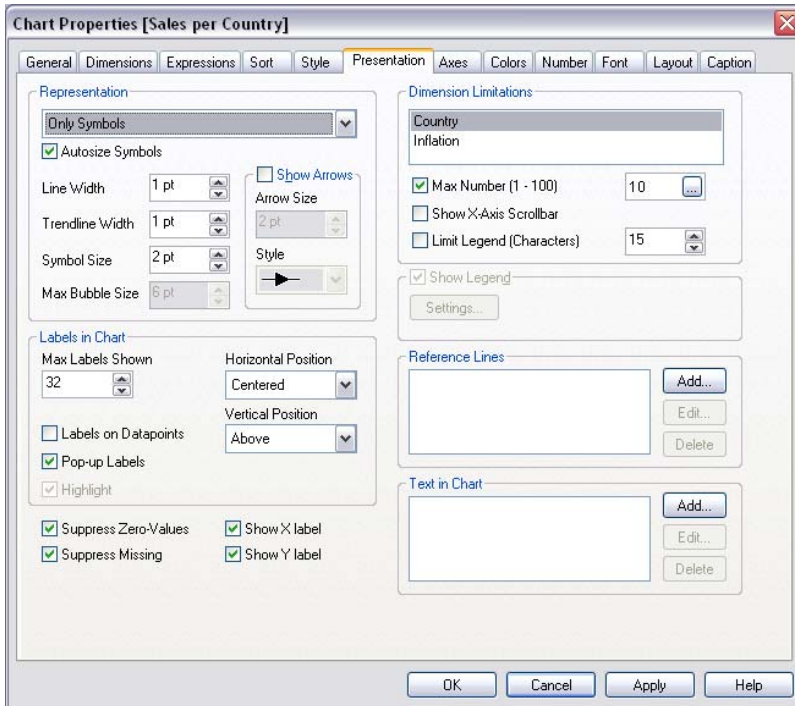


Figure 38. The Presentation page for grid charts in the Chart Properties dialog

By changing the settings on this page you modify the way the grid chart is presented on the screen.

Representation

In the **Representation** group you can set display options for plot symbols in the chart.

Autosize Symbols

If this check box is marked, the maximum size of the plot symbols in bubble mode will be automatically calculated. If unchecked the maximum size can be set manually under **Max Bubble Size** below. This setting has no meaning when uni-size plot symbols have been selected on the **Style** page.

Line Width

Sets the line width for lines. This option is available only when lines are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, px1, pixel), points (pt, pts, point)

or docunits (du, docunit). The line can be taken away by setting the width to 0.

Symbol Size

Sets the size of the symbols, when such are used. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

Max Bubble Size

Sets the size of the largest bubble in the chart when bubble style is used. All smaller values from the Z expression will be plotted smaller in a linear relation. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit). The bubbles can be taken away by setting the width to 0. This setting has no meaning when uni-size plot symbols have been selected on the **Style** page.

Labels in Chart

In the **Labels in Chart** group you can set options for showing data point labels in the chart plotting area. The labels plotted are the same as those in the legend.

Labels on Data Points

Shows labels next to the scatter points inside the chart plot area.

Pop-up Labels

By selecting this check box the value(s) of the dimension(s) will appear as a pop-up balloon when hovering with the cursor over data point in the chart. This setting works independently from the **Labels on Data Points** option. The settings below in this group have no effect on pop-up labels.

Highlight

If this check box is marked hovering with the mouse over a line or symbol in the chart will result in the line or symbol being highlighted for better overview. The highlighting applies also in the legend where applicable.

Max Labels Shown

Sets a limit for how many labels that are plotted in the chart. Setting this number too high may degrade chart readability.

Horizontal Position

In the drop-down box you can select whether the labels should be plotted to the left of the scatter point, to the right

of the scatter point or be horizontally centered on the scatter point.

Vertical Position

In the drop-down box you can select whether the labels should be plotted on top of the scatter point, below the scatter point or be vertically centered on the scatter point.

Suppress Zero Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Show X Label

Shows field label along the x-axis, instead of at the end of the axis.

Show Y Label

Shows field label along the y-axis, instead of at the end of the axis.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Max Number

Select this check box to limit the number of dimension values to be shown at any one time. Enter the maximum number of values in the scroll box.

Show X-Axis Scrollbar

Select this check box to show a scroll control in the place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values to be shown. The number of values shown at any one time will be the number set under **Max Number**. This option is only available when **Max Number** is selected.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend.

Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the chart plot area from a given point on the y-axis. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 139 for details) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click on this button in order to edit its properties in the **Reference Lines** dialog (see page 139 for details). Double-clicking on the reference line in the list has the same effect.

Delete

Mark an existing reference line in the list and click on this button in order to delete it from the list.

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog (See page 64 for details) where you may create and edit free-floating texts to be displayed in the chart.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog (See page 64 for details). Double-clicking the text in the list has the same effect.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Show Legend

Select this check box if you want legend to be shown in the chart when applicable.

Settings...

Opens the **Legend Settings** dialog (see page 60 for details), where advanced settings can be made for the chart legend.

Axes

In the **Axes** page, the display properties of the x- and y-axes settings are set. This page is identical with the **Axes** page of the scatter chart (Figure 35 on page 141).

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the grid chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).



8 PIE CHART

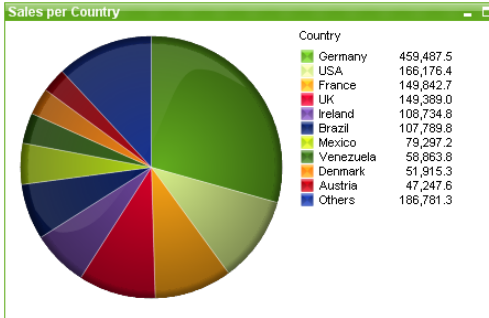


Figure 39. An example of a pie chart

Pie charts normally show the relation between a single dimension and a single expression, but can sometimes have two dimensions.

8.1 Creation

Pie charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.

If you want to create a simple pie chart and find the great number of different settings available overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu (**Quick Chart Wizard**), or click the **QuickChart** button in the toolbar. The wizard appearing will help you to create a chart in a quick and easy way. For more information, see page 265.

8.2 Pie Chart Object Menu

The pie chart **Object** menu is found as the **Object** menu when a pie chart is active. It can also be opened as a context menu by clicking with the right mouse button on a pie chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the chart static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Deselects the values of all the fields in the grid chart.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the chart.

8.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). The Pie chart cannot have more than two dimensions, any additional dimensions will be disregarded.

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39).

Note The specific use of the following setting is however possible.

Values on Data Point

Mark this check box to have the result of the expression plotted as text next to each pie slice. By creating an extra expression containing the name of the dimension field and marking it only for use as **Values on Data Point**, it is possible to show the data labels next to the pie slices.

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

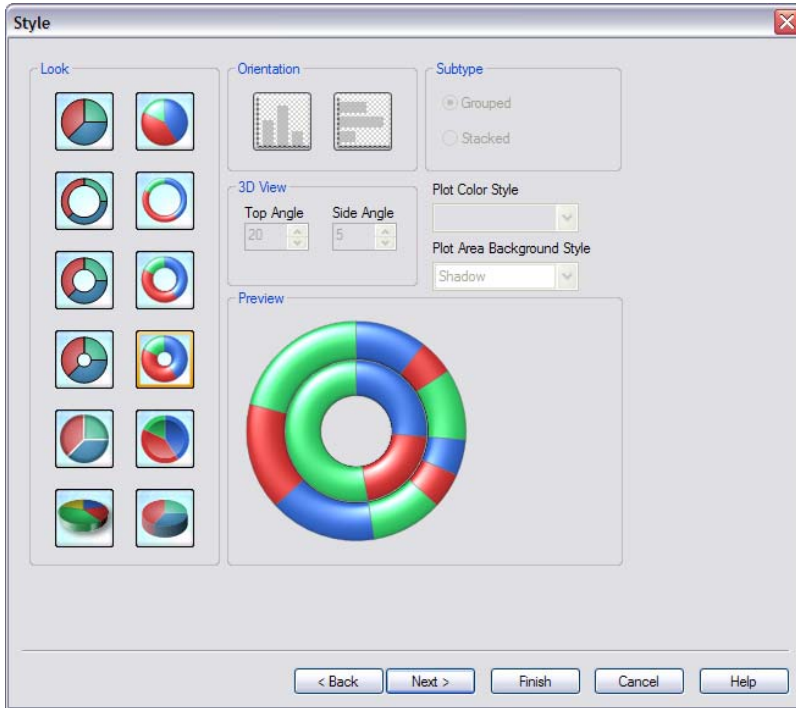


Figure 40. The Style page for pie charts in the Chart properties dialog

In this page you select the basic style of the pie chart. The options of the **Orientation**, **Subtype** and **Plot Area Background Style** groups are not available for pie charts.

Look

Select one of the available styles.

3D View

The settings in this group define the angle from which the chart is viewed in 3D modes.

Top Angle

Defines the vertical angle of the 3D view. The value must be an integer between 0 and 30.

Side Angle

Defines the side angle of the 3D view. The value must be an integer between 0 and 45.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors page** will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all segments.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

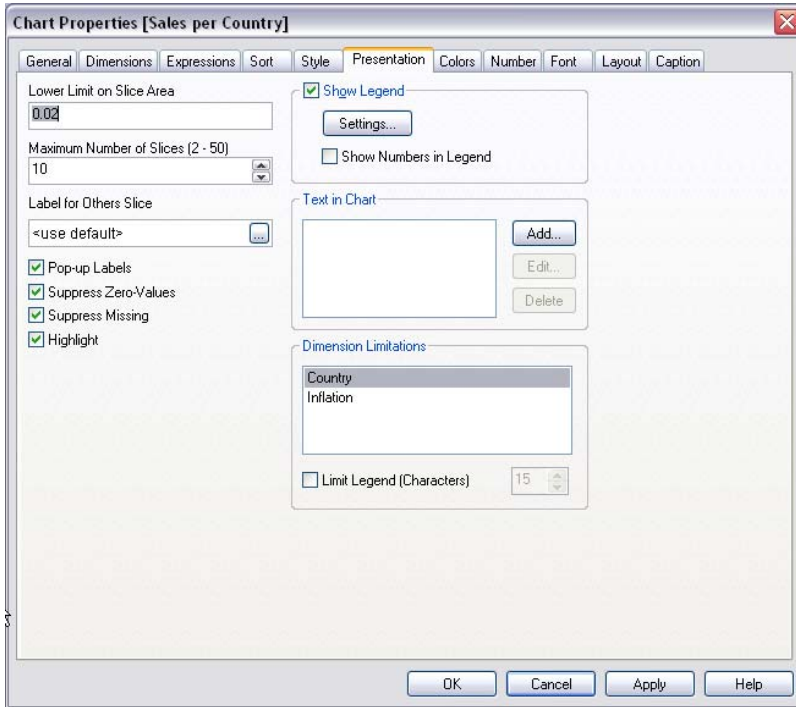


Figure 41. The Presentation page for pie charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the pie chart is presented on the screen.

Lower Limit on Slice Area

Sets the minimum size for a slice in order to be shown.

Maximum Number of Slices (2-50)

Sets the maximum number of slices.

Label for Others Slice

The name of the surplus slice. If no string is entered, the default value found in the **Objects** page of the **User Preferences** dialog will be used.

Pop-up Labels

By selecting this check box the values of the dimension(s) and expression(s) will appear as a pop-up balloon when hovering with the cursor over data point in the chart.

Suppress Zero-Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Highlight

If this check box is marked hovering with the mouse over a pie segment will result in the segment being highlighted for better overview. The highlighting applies also in the legend where applicable.

Show Legend

In the **Show Legend** group you can turn legend on or off and access the **Legend Settings** dialog. These settings are only available when chart data makes it possible to show a legend.

Settings...

Opens the **Legend Settings** dialog (page 60 for details), where advanced settings can be made for the chart legend.

Show Numbers in Legend

Shows numbers for the slices in legend. This option is only available for the main dimension in two-dimensional charts.

Text in Chart

In the **Text in Chart** group you can add free-floating text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog where you may create and edit free-floating texts to be displayed in the chart. See page 64 for details about the **Text in Chart** dialog.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog. Double-clicking the text in the list has the same effect. See page 64 for details about the **Text in Chart** dialog.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the pie chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).



9 FUNNEL CHART

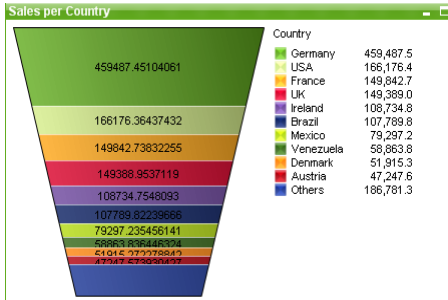


Figure 42. An example of a Funnel chart

The funnel chart is typically used for showing data in flows and processes. From a display standpoint it is related to the pie chart. The chart may be shown with either segment height/width or segment area proportional to data. It is also possible to draw the chart with equal segment heights/widths, disregarding data points.

9.1 Creation

Funnel charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart.

9.2 Funnel Chart Object Menu

The **Funnel chart Object** menu is found as the **Object** menu when a Funnel chart is active. It can also be opened as a context menu by clicking with the right mouse button on a Funnel chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the chart static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Clears all selections in the fields that are used as dimensions in the chart.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikView PDF printer pre-selected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the chart.

9.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the chart wizard is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The wizard and the **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). A funnel chart typically has one to three dimensions. Additional dimensions are ignored.

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39). In a funnel chart only the first expression is used when plotting the chart. Additional expressions can be used for pop-up purposes or as text/numbers in data segments.

Sort

On this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

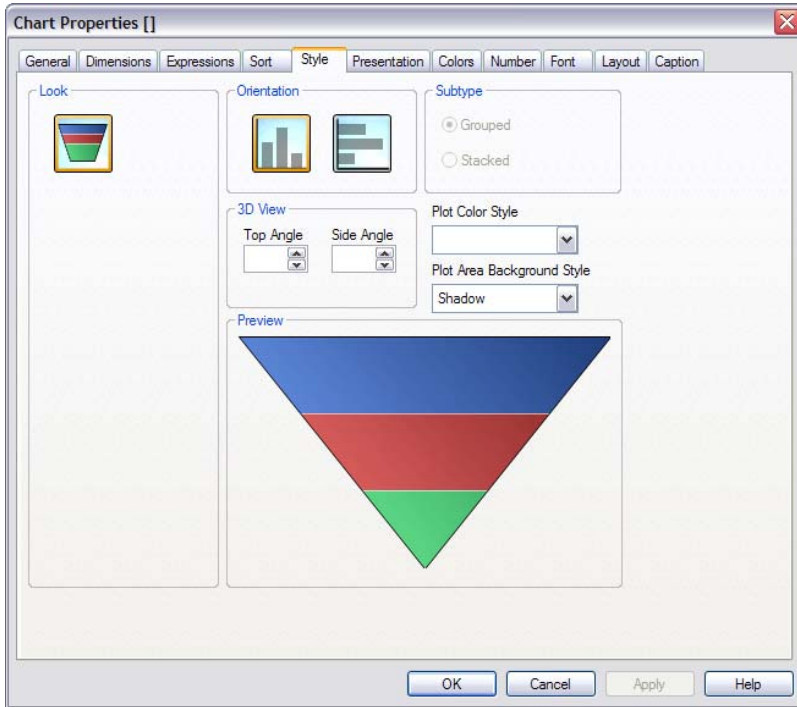


Figure 43. The Style page for Funnel charts in the Chart Properties dialog

On this page you set the style of the Funnel chart. The options of the **Subtype** group are not available to funnel charts.

Look

There is currently only one style available for funnel charts.

Orientation

The chart can be drawn in a vertical or horizontal direction.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors page** will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all segments.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

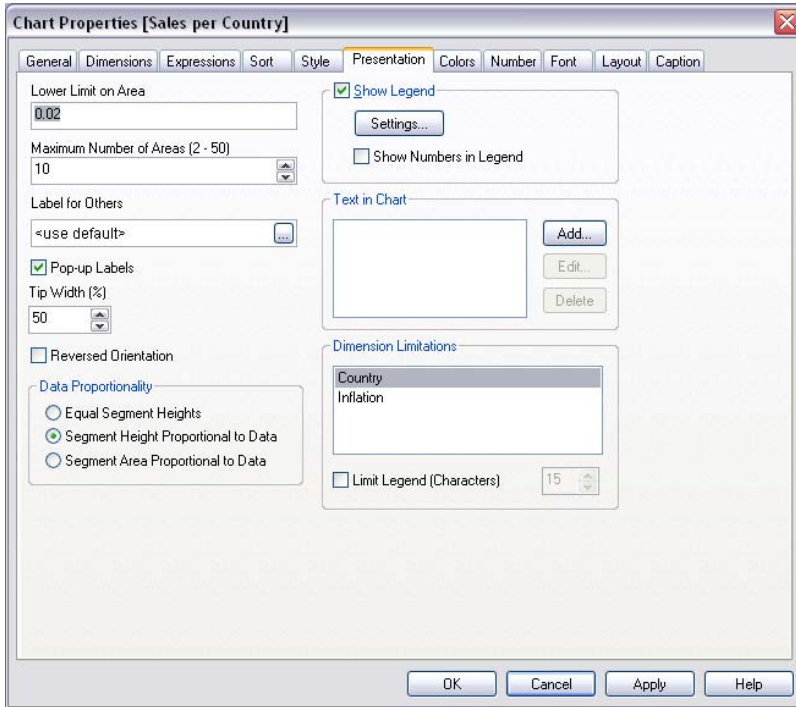


Figure 44. The Presentation page for Funnel charts

By changing the settings on this page, you modify the way the funnel chart is presented on the screen.

Lower Limit on Area

Sets the minimum size for a data segment in order to be shown. Segments that are too small to be shown will be grouped under **Others**.

Maximum Number of Areas (2-50)

Sets the maximum number of segments. Segments not shown will be grouped under **Others**.

Label for Others

The name of the surplus segment containing data for segments not shown due to limitations in the two settings above. If no string is entered, the default value found in the **Objects** page of the **User Preferences** dialog will be used.

Pop-up Labels

Mark this check box in order for data to appear in a pop-up balloon when hovering with the cursor over a data segment in the chart. This setting has no effect on expressions marked for display as pop-up.

Tip Width (%)

Defines the width of the funnel tip as a percentage of the funnel mouth width.

Reverse Orientation

Mark this check box if you want to make the funnel point to the left or upwards.

Data Proportionality

This setting controls the degree to which the segments are proportional to the underlying data.

Equal Segment Heights

No proportionality to data. Each segment is drawn with the same height (width for horizontal funnels) as all others. Typically only makes sense with numbers displayed in the chart.

Segment Height Proportional to Data

Each segment is drawn with the height of the segment proportional to underlying data.

Segment Area Proportional to Data

Each segment is drawn with the total area proportional to underlying data.

Show Legend

In the **Show Legend** group you can control the display of dimension data labels in the chart. Mark the check box in order to display data labels. Data labels are shown only for the current top level of the chart.

Settings...

Opens the **Legend Settings** dialog (see page 60 for details), where the legend settings can be modified.

Show Numbers in Legend

Mark this check box to show values next to the dimension data labels.

Text in Chart

In the **Text in Chart** group you can add free-floating text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog where you may create and edit free-floating texts to be displayed in the chart. See page 64 for details about the **Text in Chart** dialog.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog. Double-clicking the text in the list has the same effect. See page 64 for details on the **Text in Chart** dialog.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the funnel chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77)

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).

10 BLOCK CHART

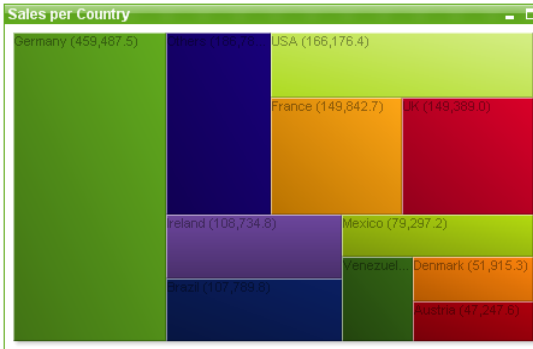


Figure 45. An example of a Block Chart

The Block chart shows the relation between expression values as blocks of varying area. Up to three dimensions can be shown, where each dimension is sub-divided in sub-blocks. An extra expression is often used to calculate the color of each block, in which case the chart type is also known as a heat chart.

10.1 Creation

Block charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** that will help you create the chart.

10.2 Block Chart Object Menu

The Block chart **Object** menu is found as the **Object** menu when a Block chart is active. It can also be opened as a context menu by clicking with the right mouse button on a Block chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the chart static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Clears all selections in the fields used as dimensions in the chart.

Print...

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the chart.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from QlikTech's homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the chart.

10.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Create Chart** wizard is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.

Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button.

The wizard and the **Chart Properties** dialog contain a number of pages which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). A Block chart typically has one to three dimensions. Further dimensions are ignored.

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39). In a Block chart only the first expression is used when plotting the chart. Additional expressions can be used for pop-up purposes. It is also possible to use **Text on Data Point** expressions for display of data in the innermost level data blocks.

Style

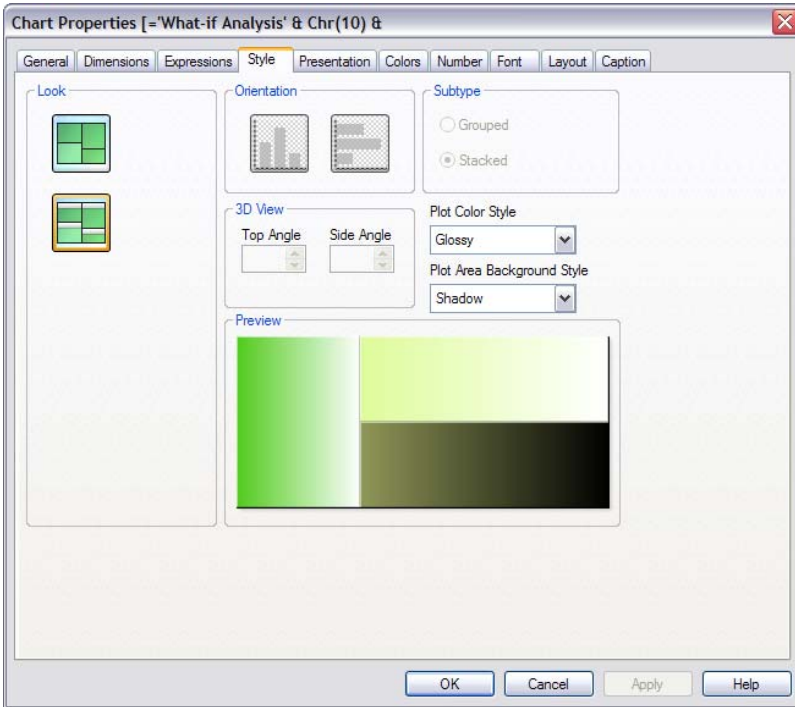


Figure 46. The Style page for Block charts in the Chart Properties dialog

In this page you set the style of the Block chart.

Look

Choose between non-captioned and captioned drawing style. In the non-captioned style only top level dimension labels will be shown in the chart, plotted in the center of their respective blocks. In the captioned style top and intermediate level dimension labels are plotted in captions at the top of their respective blocks. Bottom level dimension labels are then plotted inside each low level block.

Plot Color Style

This control can be used to impose a color style on all plot colors in the chart. When a style is selected in the drop-down all colors under **Color Map** on the **Colors** page will be changed to the selected style. The actual base colors in the color map are not affected. The **Plot Color Style** setting is not available for all chart looks. Four options are available.

Solid Color

Sets all colors in the color map to solid colors.

Dark Gradient

Sets all colors in the color map to a one-color gradient going towards a darker tone.

Light Gradient

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

Glossy

Gives a glossy look to all segments.

Plot Area Background Style

This control can be used to change the appearance of the plot area background. This setting is only available for charts with a plot area. The following options are available:

Frame

A frame is drawn around the plot area.

Shadow

This options gives a shadow effect on the plot area background.

Minimal

The plot area background is removed.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

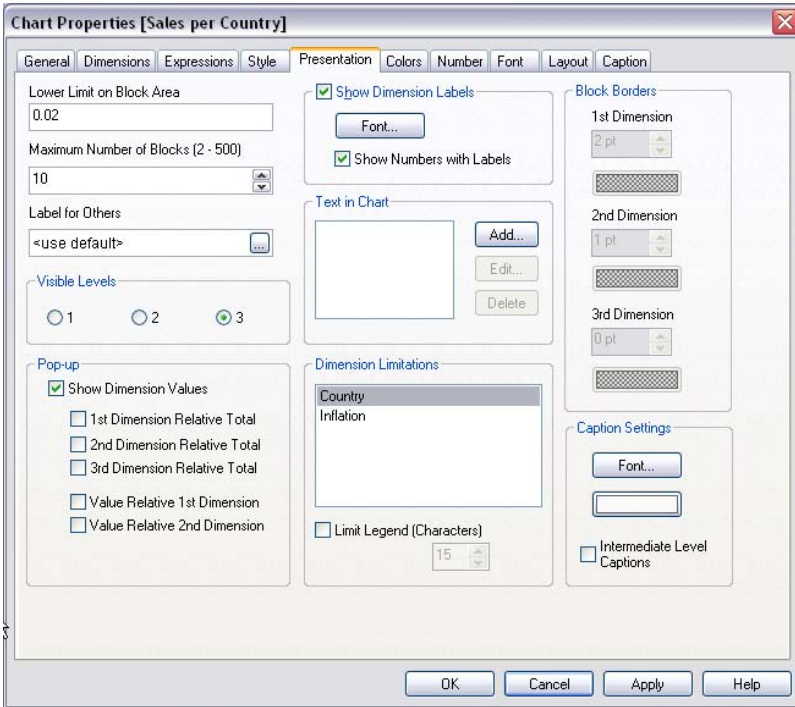


Figure 47. The Presentation page for Block charts in the Chart Properties dialog

By changing the settings on this page you modify the way the pie chart is presented on the screen.

Lower Limit on Block Area

Sets the minimum size for a block in order to be shown. Blocks that are too small to be shown will be grouped under **Others**.

Maximum Number of Blocks (2-50)

Sets the maximum number of blocks. Blocks not shown will be grouped under **Others**.

Label for Others

The name of the surplus block containing data for blocks not shown due to limitations in the two settings above. If no string is entered, the default value found in the **Objects** page of the **User Preferences** dialog will be used.

Visible Levels

A block chart can be shown with a maximum of three levels of blocks (dimensions). In this control you may limit the number of shown levels to more than one or two.

Pop-up

In this group you can specify to which extent the values of the dimension(s) and some other data will appear in the pop-up balloon when hovering with the cursor over data point in the chart. These settings have no effect on expressions marked for display as pop-up.

Show Dimension Values

Mark this check box in order to show the values of all dimensions associated with a given data point in the hover pop-up balloon. This check box must be marked in order for the options below to become available.

1st Dimension Relative Total

Mark this check box in order to include in the hover pop-up balloon the value of the chart's plot expression aggregated over the 1st dimension value as a percentage of the overall chart total.

2nd Dimension Relative Total

Mark this check box in order to include in the hover pop-up balloon the value of the chart's plot expression aggregated over the 2nd dimension value as a percentage of the overall chart total.

3rd Dimension Relative Total

Mark this check box in order to include in the hover pop-up balloon the value of the chart's plot expression aggregated over the 3rd dimension value as a percentage of the overall chart total.

Value Relative 1st Dimension

Mark this check box in order to include in the hover pop-up balloon the value of the chart's plot expression aggregated over the 1st dimension value.

Value Relative 2nd Dimension

Mark this check box in order to include in the hover pop-up balloon the value of the chart's plot expression aggregated over the 2nd dimension value.

Show Dimension Labels

In the Show Dimension Labels group you can control the display of dimension data labels in the chart. Mark the check box in order to

display data labels. Data labels are shown only for the current top level of the chart.

Font...

Opens the **Font** dialog (See page 76) where the font of the data labels can be modified.

Show Numbers with Labels

Mark this check box to show values next to the dimension data labels.

Text in Chart

In the **Text in Chart** group you can add free-floating text to the chart.

Add

By pressing this button you open the **Text in Chart** dialog where you may create and edit free-floating texts to be displayed in the chart. See page 64 for details about the **Text in Chart** dialog.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog. Double-clicking the text in the list has the same effect. See page 64 for details about the **Text in Chart** dialog.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Dimension Limitations

In the **Dimension Limitations** group you may limit the number of data points to be plotted within a given dimension. Select one of the dimensions by clicking in the list before changing one or more of the settings below.

Limit Legend (Characters)

Select this check box to limit the length of the dimension value strings to be shown on axes and in the chart legend. Enter the maximum length in the scroll box. Truncated values will be followed by ... in the chart. This option can be used independently to the other options in this group.

Block Borders

In the **Block Borders** group you can set the width and color of the borders drawn around the blocks on the three different dimension

levels. Block borders are not applicable when the caption style of block charts is used.

1st Dimension

Set the width of the border surrounding blocks on the first dimension level in the scroll control and select border color with the color button below.

2nd Dimension

Set the width of the border surrounding blocks on the second dimension level in the scroll control and select border color with the color button below.

3rd Dimension

Set the width of the border surrounding blocks on the 3rd dimension level in the scroll control and select border color with the color button below.

Caption Settings

In the **Caption Settings** group you can set the font and color of the caption used for top and intermediate dimension levels. These settings are only applicable when the caption style of block charts is used.

Font...

Opens the **Font** dialog for setting caption font.

Color

Opens the **Color Area** dialog for setting caption base color.

Intermediate Level Captions

Mark this check box if you want captions drawn on the intermediate dimension level in three-dimensional block charts.

Colors

In the **Colors** page, the default display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the block chart should appear on the layout. This includes settings for shape, border and the layer where the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).



11 GAUGE CHART

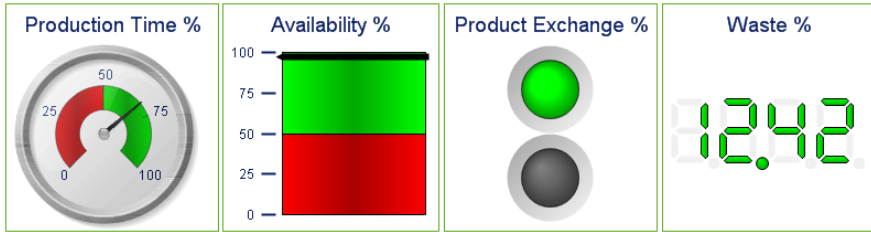



Figure 48. Four examples of gauge charts

Gauge charts are used to display the value of a single expression without dimensions.

11.1 Creation

Gauge charts are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the chart. 

If you want to create a simple gauge chart and find the great number of different settings available overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu (**New Object - QuickChart**), or click the **QuickChart** button in the toolbar. The wizard appearing will help you to create a chart in a quick and easy way. For more information, see page 265.

11.2 Gauge Chart Object Menu

The gauge chart **Object** menu is found as the **Object** menu when a gauge chart is active. It can also be opened as a context menu by clicking with the right mouse button on a gauge chart. The menu contains the following commands:

Properties

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the chart static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the chart.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Print

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the table.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a DF printer is available on the system. The QlikViewPDF printer must be downloaded from QlikTech's homepage and installed separately.

Send Values to Excel

Exports the underlying data (the straight table equivalent to the chart) to Microsoft Excel, that is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Opens a dialog for saving an image of the gauge to file. The image can be saved as bmp, jpg, gif or png.

Copy to Clipboard

This menu contains the various copy options for the chart object.

Values

Copies the values to the clipboard in the form of a table.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the chart to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the chart.

11.3 Chart Properties

When you click the **Create Chart** button in the toolbar, the chart wizard is opened. Here you can set the properties of the chart, e.g. the chart type, the dimensions, titles, etc.



Once a chart is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The wizard and the **Chart Properties** dialog contain a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the chart can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). Gauge charts do not make use of chart dimensions, so this page can be ignored.

Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical with the **Expressions** page of the bar chart (Figure 6 on page 39). The gauge chart will always display the value of the first expression only.

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Style

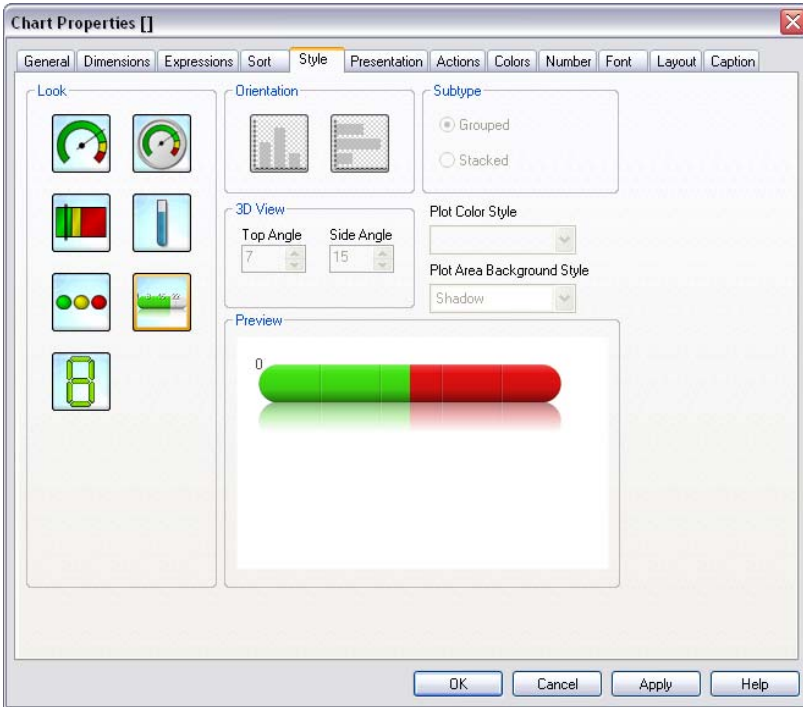


Figure 49. The Style page for gauge charts

In this page you select the basic style of the gauge chart. The options of the **3D View**, **Subtype**, **Plot Color Style** and **Plot Area Background Style** groups are not available for gauge charts.

Look

Select one of the available styles.

Orientation

Here you can set the orientation of the chart to either vertical or horizontal as indicated by the icons. The setting is only relevant for linear and traffic light style gauges.

Preview

Offers a preview of the basic visual properties of the chart.

Presentation

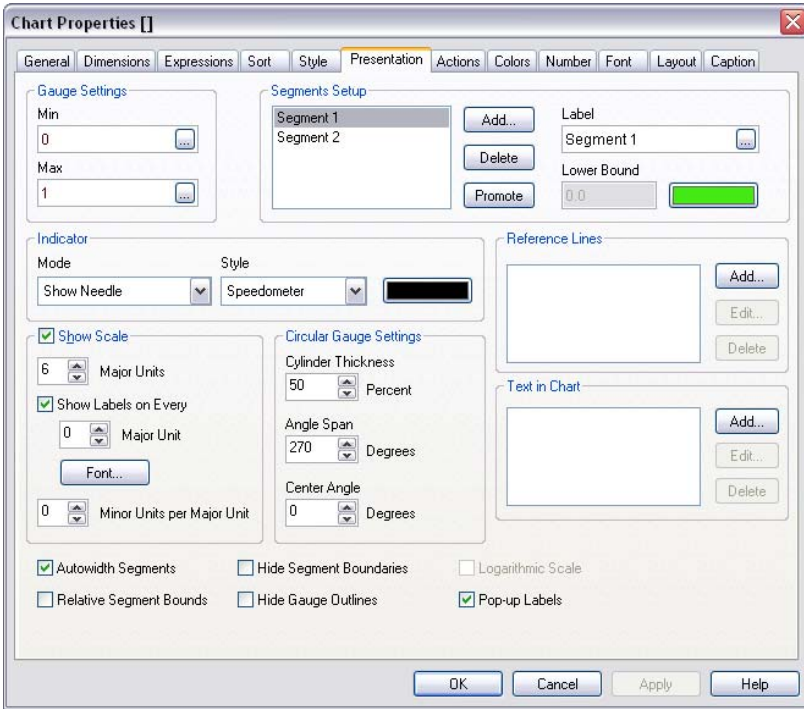


Figure 50. The Presentation page for circular gauge charts in the Chart Properties dialog

By changing the settings on this page, you modify the way the gauge chart is presented on the screen. The **Presentation** page gauge charts looks slightly different depending on which style you have selected on the **Style** page.

Gauge Settings

In the **Gauge Settings** group you set minimum and maximum values for the gauge chart.

Min

Specifies the minimum value of the gauge. This corresponds to the bottom position of the gauge indicator.

Max

Specifies the maximum value of the gauge. This corresponds to the top position of the gauge indicator.

Segments Setup

In this group you define the segments making up the gauge. All gauges except LED style gauges must have at least one segment. For circular and linear gauges the segments make up differently colored areas in the gauge background. For traffic light gauges each segment corresponds to one light. A list of segments is shown and you may select one segment in the list to change its properties.

Add

Adds a new segment at the end of the list.

Delete

Deletes the segment currently selected in the list.

Promote

Moves the segment currently selected in the list one step up.

Label

Specifies a name for the selected segment. This name is for identification only and is not used when drawing the gauge chart.

Lower Bound

Specifies the gauge value where the selected segment starts. The upper bound of a segment is the lower bound of the following segment or, in case of the last segment the gauge's **Max** value. This value can only be edited if the option **AutoWidth Segments** (see below) has been deselected. The value can be entered as a formula for dynamic update. If the option **Relative Segment Bounds** (see below) has been selected the segment bounds should be entered as numbers between 0 and 1 indicating a fraction of the total range between the **Min** and **Max** values.

Color

Click the color button to specify the color of the segment. The color can be defined as a solid color or a gradient via the **Color Area dialog** (see page 23) that opens when clicking the button. Gradients have no effect in traffic light style gauges.

Indicator

In the **Indicator** group you set the visual properties of the gauge indicator. This group is not available for LED style gauge charts.

Mode

Specifies the value display mode of the gauge.

For circular and linear style gauges the following four modes are available:

Show Needle

Shows the gauge value by means of a needle indicator. The type of needle can be set under **Style** below.

Fill to Value

Shows the gauge value by means of filling the gauge background with the defined segment color(s) only up to the current value. The remaining part up to the **Max** value is left blank.

Needle and Fill to Value

Combines the two options above.

No Indicator

Neither needle indicator nor fill to value is used. This option can be valuable when using dynamically calculated segment bounds to indicate gauge value(s).

For traffic light style gauges the following three modes are available:

Indicate Value

Shows the gauge value by lighting up the single light corresponding to the current gauge value.

Fill to Value

Shows the gauge by lighting up the lights corresponding to all segments up until and including the segment containing the current gauge value.

Single Light

Shows the gauge value by lighting up the single light corresponding to the segment containing the current gauge value while hiding all other segment lights. Creates the effect of one single light changing colors depending on the color of the segment corresponding to the current gauge value.

Style

Specifies the style of the needle for circular and linear style gauges and for the area surrounding the lights in traffic

light style gauges. Several styles are available depending on the type of gauge.

Color

Click the color button to specify the color of the indicator needle. Not applicable for traffic light gauges.

Reverse Direction

Select this check box to reverse the plot direction of a traffic light gauge. This option is only available for traffic light gauges.

Show Scale

Select this check box to show a scale. This group is only available for circular and linear style gauges.

Major Units

Defines the number of major units on the gauge scale.

Show Labels on Every n Major Unit

Select this check box to show text labels on the gauge scale. The density of labels can be entered in the edit box.

Font

Press the button to set a font for the scale label text.

Minor Units per Major Unit

Defines the number of minor units between each major unit on the gauge scale.

Circular Gauge Settings

In this group, which is specific to circular style gauge charts, you specify the properties of the gauge shape.

Cylinder Thickness

The gauge is normally drawn as a solid circle or circle segment. The greater the number here, the thicker the cylinder. The value which indicates the percentage of the radius to be left unfilled must be between 0 and 99.

Angle Span

The angle in degrees between the **Min** and **Max** values in the gauge. Must be a value between 45 and 360.

Center Angle

The angle of the gauge's center value in relation to the clock. Must be a value between 0 and 360. 0 indicates the center at the top of the gauge (12 o'clock).

Text in Chart

In the **Text in Chart** group you can add free text to the chart.

Add

By pressing this button you open the **Text in Chart** button where you may create and edit free-floating texts to be displayed in the chart. See page 64 for details about the **Text in Chart** dialog.

Edit

Mark an existing text in the list and click this button in order to edit its properties in the **Text in Chart** dialog.

Delete

Mark an existing text in the list and click this button in order to delete it from the list.

Reference Lines

In the **Reference Lines** group you can define reference lines intersecting the gauge background area at a given value. Reference lines are only available in circular and linear style gauges. A list shows all existing reference lines in the chart.

Add

Opens the **Reference Lines** dialog (see page 62 for details) for creation of a new reference line in the chart.

Edit

Mark an existing reference line in the list and click this button in order to edit its properties in the **Reference Lines** dialog. Double-clicking the reference line in the list has the same effect. See page 62 for details about the **Reference Lines** dialog.

Delete

Mark an existing reference line in the list and click this button in order to delete it from the list.

Autowidth

If this check box is marked, the segment bounds will be calculated automatically based on the gauges **Min** value, **Max** value and the number of segments defined.

Relative Segment Bounds

If this option is selected the segment bounds can be entered as numbers between 0 and 1 indicating a fraction of the total range between the **Min** and **Max** values.

Hide Segment Boundaries

If this check box is marked no outline will be made in the segment boundaries of circular and linear gauges. Useful when creating background gradients with more than two colors.

Hide Gauges Outline

Removes the outline of the gauge segments.

Logarithmic Scale

If this check box is marked, the gauge scale will be logarithmic.

Pop-up Labels

By selecting this check box the value expression(s) will appear as a pop-up balloon when hovering with the cursor over data points in the chart.

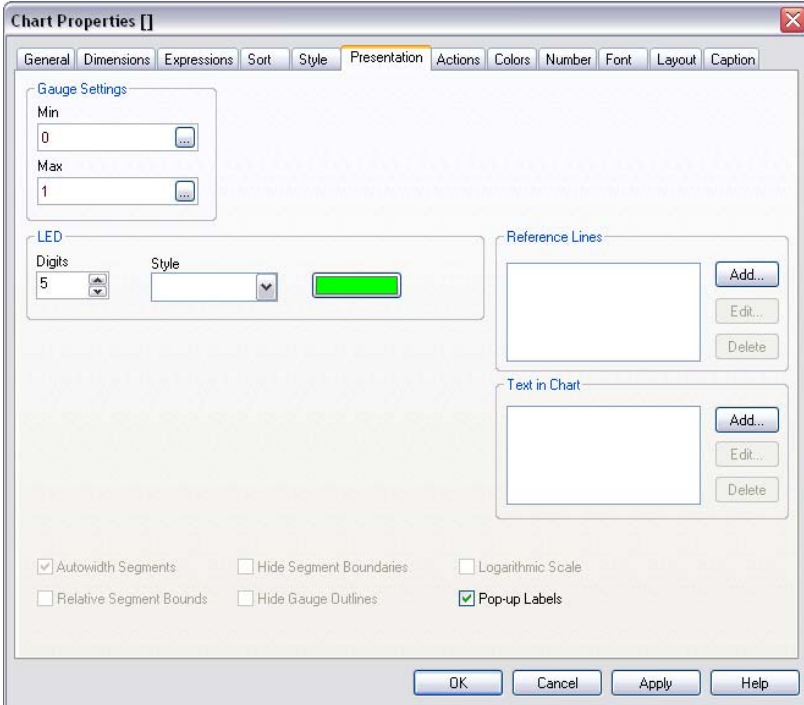


Figure 51. The Presentation page for LED gauge charts in the Chart Properties dialog

LED

In the LED group, which is specific to LED style gauge charts, you set the visual properties for the LED display.

Digits

Specifies the number of LED display digits shown.

Color

Click the color button to specify the color of the LED segments.

Style

Choose a style for the LEDs in this drop-down menu.

Actions

In the **Actions** page you can specify what actions should be performed when you click in the chart. The page is identical to the **Actions** page for the **Button Object**, see page 200 in Book II.

Colors

In the **Colors** page, the display colors can be set. This page is identical with the **Colors** page of the bar chart (Figure 14 on page 70). Note that the color map under **Data Appearance** has no meaning in a gauge chart.

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the chart. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76).

Layout

In the **Layout** page, you can specify how the gauge chart should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).

12 PIVOT TABLE

Sales per CategoryName			
CategoryName	ProductName	Sales	Quantity
Men's Clothes	Atlas Lussekkotta	\$30,126.55	1057
	Blow tie	\$9,534.57	1315
	Desperado Jeans	\$18,240.68	706
	Lenin Jeansshorts	\$14,900.64	628
	Mr2 Trousers	\$17,944.48	1067
	O-Man Underwear	\$1,649.87	298
	Rossi Bermuda Shorts	\$10,847.25	1397
	Samba Soccer Socks	\$4,941.14	1175
	US-Master Jeans	\$21,784.94	817
	Total	\$130,059.12	8660
Women's Clothes	Chantel Shirt	\$7,504.70	388
	Heater Dress	\$361,096.85	381
	Jack Flash Dress	\$42,639.00	722
	Langeste Shirt	\$4,433.35	246
	Le Baby Dress	\$47,571.88	623
	Minniki Palsi	\$10,472.71	184
	Okkaba Skin Jackets	\$42,258.78	601
Oyaki Kimono	\$9,084.42	806	

Figure 52. An example of a pivot table

The pivot table is one of the most powerful tools for analyzing data. It offers substantial functionality but is still easy to use. Pivot tables show dimensions and expressions in rows and columns, for example in cross tables. The data in pivot tables may be grouped. Pivot tables can show partial sums.

12.1 Creation

Pivot tables are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the table. Pivot tables are logically equivalent to charts.

If you want to create a simple pivot table and find the great number of different settings available overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu (**QuickChart**), or click the **QuickChart** button in the toolbar. The wizard appearing will help you to create a table in a quick and easy way. For more information, see page 265.

12.2 Using the Pivot Table

Pivoting - Moving the Dimension Fields

In a pivot table, the dimension fields and expressions can be shown on one vertical and one horizontal axis. Dimensions and expressions can be moved freely between or within the axes. This is called pivoting. Pivoting in QlikView is done by means of dragging and dropping with the mouse.

Note You can disable the pivoting of a pivot table by deselecting the check box **Allow Pivoting** on the **Presentation** page in the **Properties** dialog of the pivot table.

Example:

To move the field *Product* in the pivot table below from the vertical axis to the horizontal axis, point with the mouse in the *Product* column. Press the left mouse button and keep it depressed while dragging the mouse cursor up above the expression label row. The selected column and its target are highlighted in blue while you are dragging.

sum(Sales)				
Company	Product	Year	sum(Sales)	avg(Sales)
ABC	A	1998	3	3.0
		1999	2	2.0
		Total	5	2.5
	B	1998	1	1.0
		1999	1	1.0
		Total	2	1.0
Total			7	1.8
XYZ	A	1998	5	5.0
		1999	4	4.0
		Total	9	4.5
	B	1998	7	7.0
		1999	6	6.0
		Total	13	6.5
Total			22	5.5
Total			29	3.6

The result:

sum(Sales)							
Company	Product	A		B		Total	
	Year	sum(Sales)	avg(Sales)	sum(Sales)	avg(Sales)	sum(Sales)	avg(Sales)
ABC	1998	3	3.0	1	1.0	4	2.0
	1999	2	2.0	1	1.0	3	1.5
	Total	5	2.5	2	1.0	7	1.8
	Total						7
XYZ	1998	5	5.0	7	7.0	12	6.0
	1999	4	4.0	6	6.0	10	5.0
	Total	9	4.5	13	6.5	22	5.5
	Total						22
Total		14	3.5	15	3.8	29	3.6

Not only the dimension fields, but also the expression row can be pivoted:

sum(Sales)		A		B		Total	
Company	Year	sum(Sales)	avg(Sales)	sum(Sales)	avg(Sales)	sum(Sales)	avg(Sales)
ABC	1998	3	3.0	1	1.0	4	2.0
	1999	2	2.0	1	1.0	3	1.5
	Total	5	2.5	2	1.0	7	1.8
XYZ	1998	5	5.0	7	7.0	12	6.0
	1999	4	4.0	6	6.0	10	5.0
	Total	9	4.5	13	6.5	22	5.5
Total		14	3.5	15	3.8	29	3.6

The result:


sum(Sales)		Product	A	B	Total
ABC	1998	sum(Sales)	3	1	4
		avg(Sales)	3.0	1.0	2.0
	1999	sum(Sales)	2	1	3
		avg(Sales)	2.0	1.0	1.5
Total	sum(Sales)	5	2	7	
	avg(Sales)	2.5	1.0	1.8	
XYZ	1998	sum(Sales)	5	7	12
		avg(Sales)	5.0	7.0	6.0
	1999	sum(Sales)	4	6	10
		avg(Sales)	4.0	6.0	5.0
Total	sum(Sales)	9	13	22	
	avg(Sales)	4.5	6.5	5.5	
Total		sum(Sales)	14	15	29
		avg(Sales)	3.5	3.8	3.6




Expanding and Collapsing the Dimension Axes

QlikView pivot tables allow you to expand and collapse dimensions on the axes by single field values. This allows you to drill down into details for one or more field values while keeping the totals for others.

Note You can disable the expand and collapse options for a pivot table by selecting the check box **Always fully expanded** on the **Presentation** page in the pivot table **Properties** dialog.

Example (expanding):

In the pivot table below, the  icons to the right of the field values in the Company column indicate that the table can be expanded for further detail.


sum(Sales)		sum(Sales)	avg(Sales)
Company			
ABC		7	1.8
XYZ		22	5.5
Total		29	3.6

A click on the first  displays the Product details for the company ABC:

sum(Sales)			
Company	Product	sum(Sales)	avg(Sales)
ABC	A	5	2.5
	B	2	1.0
	Total	7	1.8
XYZ		22	5.5
Total		29	3.6


Each field value can be separately expanded to show the next level. In order to expand all field values in a certain column, right-click in that column and select **Expand All** from the object menu. If you do so for the Company column in the table above, the result will be as follows:

sum(Sales)			
Company	Product	sum(Sales)	avg(Sales)
ABC	A	5	2.5
	B	2	1.0
	Total	7	1.8
XYZ	A	9	4.5
	B	13	6.5
	Total	22	5.5
Total		29	3.6



The  icons in the Product column indicate that another level exists.

Right-click in the Product column and select **Expand All** again. The result will be:

sum(Sales)				
Company	Product	Year	sum(Sales)	avg(Sales)
ABC	A	1998	3	3.0
		1999	2	2.0
		Total	5	2.5
	B	1998	1	1.0
		1999	1	1.0
		Total	2	1.0
Total		7	1.8	
XYZ	A	1998	5	5.0
		1999	4	4.0
		Total	9	4.5
	B	1998	7	7.0
		1999	6	6.0
		Total	13	6.5
Total		22	5.5	
Total		29	3.6	

As no  icons are available in the Year column, we can conclude that there are only three dimension fields available in this pivot table.

Example (collapsing):

Just as you expand with the  icons, you can collapse individual values by clicking on the  icons.

If you click on the  icon to the right of the value A in the table above, the result will be:

sum(Sales)				
Company	Product	Year	sum(Sales)	avg(Sales)
ABC	A		5	2.5
	B	1998	1	1.0
		1999	1	1.0
		Total	2	1.0
	Total		7	1.8
XYZ	A		9	4.5
	B	1998	7	7.0
		1999	6	6.0
		Total	13	6.5
	Total		22	5.5
Total		29	3.6	

Finally, even the first column can be collapsed, leaving only the grand total of the expression for the vertical axis. This is done by right-clicking in any of the field dimension columns and choosing **Collapse Dimension Columns** from the object menu. The result will be as follows:

sum(Sales)		
	sum(Sales)	avg(Sales)
	29	3.6

From here you may expand again!

The expand and collapse possibilities apply equally for multiple dimension fields on the horizontal axis as is illustrated in the pivot table below.

sum(Sales)							
Company	Product	A		B		Total	
	Year	sum(Sales)	avg(Sales)	sum(Sales)	avg(Sales)	sum(Sales)	avg(Sales)
ABC	1998	3	3.0	1	1.0	4	2.0
	1999	2	2.0	1	1.0	3	1.5
	Total	5	2.5	2	1.0	7	1.8
XYZ	1998	5	5.0	7	7.0	12	6.0
	1999	4	4.0	6	6.0	10	5.0
	Total	9	4.5	13	6.5	22	5.5
Total		14	3.5	15	3.8	29	3.6

12.3 Pivot Table Object Menu

The pivot table **Object** menu is found as the **Object** menu when a pivot table is active. It can also be opened as a context menu by clicking with the right mouse but-

ton on a pivot table. The menu contains the following commands (the commands may differ depending on what field you click on):

Properties...

Opens the **Chart Properties** dialog, from which you can set the properties of the chart. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Expand All

Expands all expandable cells.

Collapse All

Collapses all collapsable cells.

Collapse Dimension Rows

Collapses all dimension rows.

Collapse Dimension Columns

Collapses all dimension columns.

Detach

Makes the table static, i.e. cuts the hot link to the selected data.

Attach

Attaches a detached chart, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the table.

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Fit Columns to Data

Adjusts the width of all columns in the tables to the widest data in each column. The header is included in the calculation.

Equal Columnwidth

Sets the columnwidth in the table.

Custom Format Cell

Opens the **Custom Format Cell** dialog (see page 160 in Book II) which lets you format cells in the column you clicked upon. The command is only available when design grid is turned on or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I).

Change Value

Only available for expression columns containing an **inputsum** aggregation of an input field. Sets the cell that has been clicked in input edit mode. Equivalent to clicking the input icon in the cell.

Restore Values

Only available for expression columns containing an **inputsum** aggregation of an input field. Opens a menu with three options.

Restore Single Value

Restores the field values underlying the cell that has been clicked to their default values from the script.

Restore Possible Values

Restores the values of all possible underlying field values to their default values from the script.

Restore All Values

Restores the values of all field values to their default values from the script.

Clear All Selections

Deselects the values of all the dimensions in the pivot table.

Export...

Exports the contents of the table to a file of your choice. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data files, see page 497 in Book I).

Copy to Clipboard

This menu contains the various copy options for the chart object.

Full Table

Copies the table to the clipboard, complete with header and selection status.

Table Data Area

Copies only the values of the table to the clipboard.

Cell Value

Copies the text value of the list box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.

Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.

Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.

Print...

Opens the standard **Print** dialog box (see page 263 in Book II), allowing you to print the table.


Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from QlikTech's homepage and installed separately.


Send to Excel

Exports the table to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet.

Minimize


Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the table to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available

only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.


Help

Opens context specific help.

Remove

Removes the table.

12.4 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart or table, e.g. the chart type, the dimensions, titles, etc. 

Once a table is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contains a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart or table etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the table can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31). A pivot table can have a large number of dimensions. The limit is set by the RAM available.

Expressions

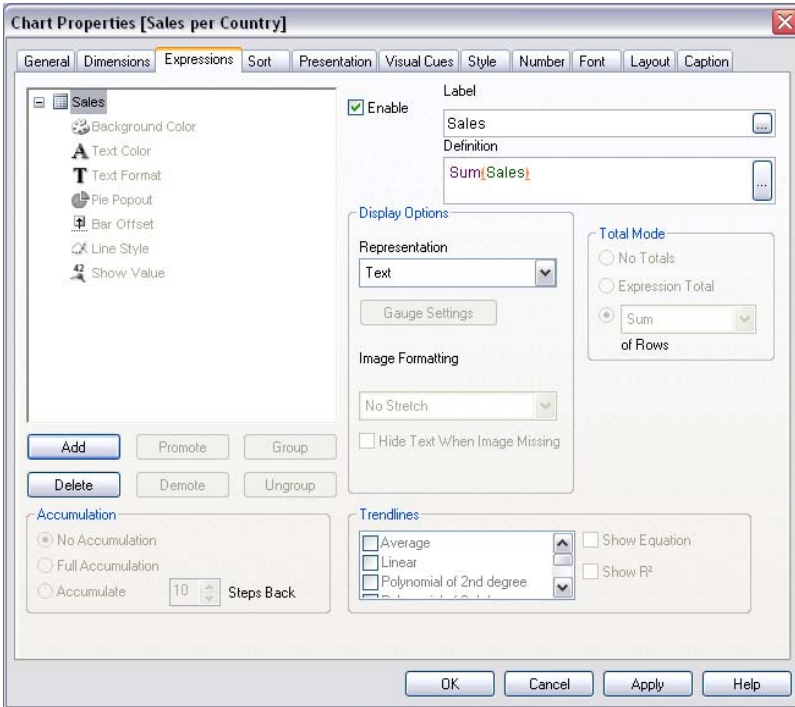




Figure 53. The Expressions page for pivot tables in the Chart Properties dialog

On the **Expressions** page you set the expressions to be displayed in the table. The Expressions page for table charts differ in a number of ways to the corresponding page for bitmap charts. As an example trendlines, accumulation, and relative expressions are not available in pivot tables.

Expressions


The upper left corner of the dialog contains a list of the chart's expressions. An expression defines the calculated contents of a chart, e.g. the data shown in an expression column in the table.

Navigating the Tree Control

The expression list is in fact a tree control with a large set of control options. In front of each expression you will find an expansion icon . By clicking the expansion icon you open up underlying sub expressions or attribute expressions. The icon will then shift to a collapse icon . By clicking that icon you collapse the display of

any sub expressions or attribute expressions. In complex cases up to three levels of expansion may be possible.


Display Options / Plot Modes

After the expansion/collapse icon on each row you will see an icon corresponding to the type of display to be used for the expression. These are normally set in the **Display Options** group (see below). There is for example one icon **T** for expressions plotted as plain text, one  for expressions plotted as gauges in cells and so on. Only icons reflecting plot types possible in the active chart type will be shown.

Sub expressions

Certain plot options in bitmap charts make use of sub expressions, i.e. a set of two or more expressions that together define the plot symbol. If **Stock** or **Box** plot has been selected as **Display Option** for an expression in a combo chart, that expression will be displayed as an empty placeholder with four or five sub expressions when changing the chart type into a table chart. The display options in the table chart can be set individually for each of the sub expressions. All expression definitions lie within the sub expressions.

Attribute expressions

Expression data can be dynamically formatted by means of attribute expressions. Click on the  expansion icon in front of any expression to display the placeholders for the dimension's attribute expressions. You may choose to use any number and combination of attribute expressions on a given base expression. Formatting defined by means of attribute expression supersedes all formatting implied by table style or visual cues. Whenever an attribute expression is entered for a dimension, its icon will turn from gray scale to color, or as in the case of **Text Format** from gray to black. The available attribute expression types are:

Background Color

Click on the **Background Color** in order to define an attribute expression for calculating the table cell background color of the data point. The calculated color will have precedence over table style and visual cues. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see

page 443). If the result of the expression is not a valid color representation, white background will be used.


A Text Color

Click **Text Color** in order to define an attribute expression for calculating the table cell text color of the data point. The calculated color will have precedence over table style and visual cues. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used.

T Text Format

Click **Text Format** in order to enter an attribute expression for calculating the font style of text in the table cell for each dimension cell. The calculated text format will have precedence over table style. The expression used as text format expression should return a string containing a B for bold text, I for italic text and/or U for underlined text.

Expression Cycle Groups

Finally there is the concept of expression groups. Any number of base expression (with their respective sub expressions and attribute expressions) may be bundled together in a group. A group is shown in the list as a placeholder with a  cycle icon. By expanding the group, you can see the expressions it contains. In the chart only the first expression in the list will be used for display. The expression to be shown out of the group can be changed by clicking the cycle icon in the chart or in the table expression column header. By right-clicking the cycle icon, you get a pop-up list of the available expressions for direct selection. When you cycle the group the second expression will be promoted to the top of the list and the previously active expression demoted to the bottom of the list.

You can add an expression to a group in two ways. The first method is by simply dragging an expression on top of another expression in the list (creates a new cycle group) or on top of an existing cycle group in the list (adds the expression to the existing group). The second method is by selecting an expression in the list and then click on the Group button (see below). The expression will then be grouped with the expression or cycle group residing above it in the list.

You can correspondingly remove an expression from a group by dragging it out of the group in the list or by selecting it and pressing the **Ungroup** button (see below). If only one expression remains in the cycle group after the extraction, that expression will also be extracted and the cycle group will cease to exist.

Add

By clicking this button, you get to the **Edit Expression** dialog (page 249), where it is possible to create new expressions.

Delete

Deletes the selected expression.

Promote

Promotes the selected expression one step up in the list. Use this button if you want to change the order of the expressions.

Demote

Demotes the selected expression one step down in the list. Use this button if you want to change the order of the expressions.

Group

It is possible to group two or more expressions, so that only one of them is displayed in the chart at any given time. The expression to be shown out of the group can be changed by clicking the cycle icon in the chart or in the table expression column header. By right-clicking the cycle icon, you get a pop-up list of the available expressions for direct selection. You may assign an expression to a cycle group by selecting an expression in the list and then click on the **Group** button. The expression will then be grouped with the expression or cycle group residing above it in the list.

Ungroup

This command is only available when an expression belonging to a cycle group has been selected in the Expressions list above. By clicking the button the expression will be moved out of the cycle group and placed as an independent expression after the cycle group in the list. If only one expression remains in the cycle group after the extraction, that expression will also be extracted and the cycle group will cease to exist.

Label

The label of the expression. Enter the name you wish to display in the chart. If no text is entered, the label will be automatically set to the expression text. The text may be entered as a calculated expression for automatic update.

Definition

Shows the composition of the selected expression. It is possible to edit the expression directly in this box. By clicking the ... button the full **Edit Expression** dialog is opened.

Enable

Marking this check box will enable the selected expression. If the box is not marked, the expression will not be used at all.

Relative

Marking this check box will set the chart to show the result in percent instead of absolute numbers. This option is not available for pivot tables.

Total Mode

This group is not available for pivot tables. By selecting one of the options in this group, you decide how the total of the selected expression is to be calculated. This setting is important for relative display or when showing totals. In pivot tables **Expression Total** will always be used and this setting will be disabled in the dialog.

No Totals

If this option is selected no total will be calculated for the expression.

Expression Total

If this option is selected, the expression total will be calculated using all the values of the field. If, e.g., the selected column contains the average salaries for different business categories, choosing the Expression Total option will result in the average salary for all the business categories.

F(x) of Rows

If this option is selected, the individual values of each data point (each bar in a bar chart, each row in a straight table etc.) for the selected expression will be summed up aggregated using the selected aggregation function (typically summed up).

Display Options

This group defines what will be written in the table chart expression cells.

Representation

The following alternatives are given:

Text

When selecting this option the expression values will always be interpreted and displayed as text.

Image

When selecting this option QlikView will try to interpret each expression value as a reference to an image. The reference may be a path to an image file on disk (e.g. `C:\MyPic.jpg`) or inside the qvw document (e.g. `qmem://<Name>/<Peter>`). If QlikView cannot interpret an expression value as a valid image reference, the value itself will be displayed.

Circular Gauge

When selecting this option QlikView will display the expression value in a circular style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

Linear Gauge

When selecting this option QlikView will display the expression value in a horizontal linear style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

Traffic Light Gauge

When selecting this option QlikView will display the expression value in a horizontal traffic light style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

LED Gauge

When selecting this option QlikView will display the expression value in a LED style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

Link

Select this option to enter an expression in the **Definition** field that will create a clickable link in the table cell. The expression should return a text that can be interpreted as *DisplayText<url>LinkText*. The *DisplayText* will be displayed in the table cell and *LinkText* will be the link that is opened in a

new browser window. If a link is defined, the value in the table cell will be underlined. If no link is defined the value will not be underlined. Note that it is not possible to make selections in a cell with **Link** as display mode. By clicking the ... button the full **Edit Expression** dialog is opened.

Examples:

```
=Name & '<url>' & Link
```

```
=Name & '<url>www.qlikview.com'
```

where *Name* and *Link* are table fields loaded in the script.

Hide

When selecting this option the expression is not shown in the table.

Gauge Settings...

Only available when one of the gauge options has been selected above. Opens a dialog for setting the properties of the gauge. This dialog is essentially equal to the **Presentation** page of Chart Properties for gauge charts (see page 194).

Image Formatting

Only available when the image options have been selected above. This setting describes how QlikView formats the image to fit in the cell. There are four alternatives.

No Stretch

If this option is selected, the image will be shown as is, without any stretching. This may cause parts of the picture to be invisible or only part of the cell to be filled.

Fill

If this option is selected, the image will be stretched to fit the cell without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the image will be stretched as far as possible to fill the cell while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the image will be stretched to fill the cell in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.

Hide Text When Image Missing

If this option is selected, QlikView will not display the field value text if interpretation as an image reference fails for some reason. The cell will then be left blank.

Sort

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical with the **Sort** page of the bar chart (Figure 7 on page 50).

Presentation

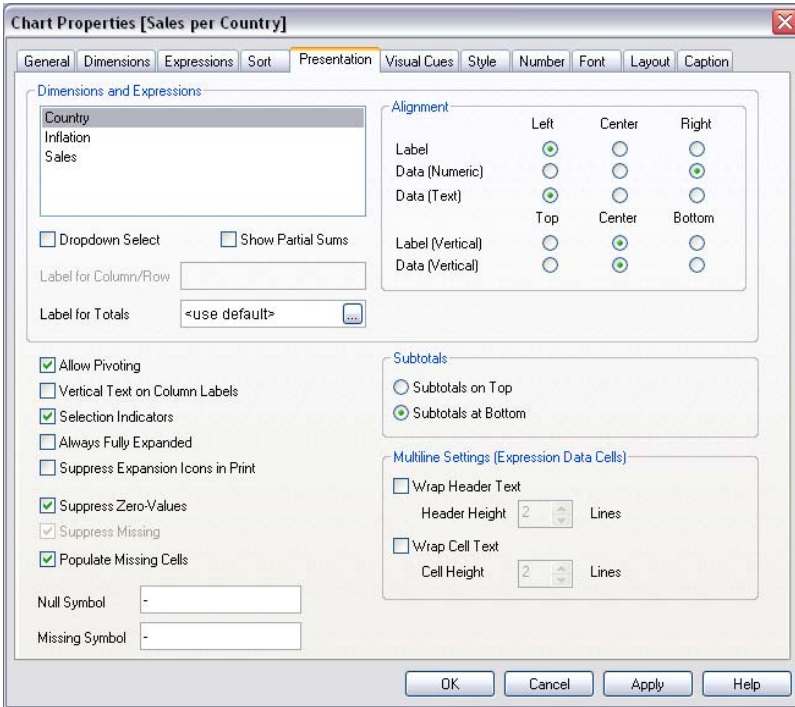


Figure 54. The Presentation page for pivot tables in the Chart Properties dialog

By changing the settings on this page, you modify the way the pivot table is presented on the screen.

Dimensions and Expressions

A list of the dimensions and expressions chosen for display in the pivot table. Selecting one from the list makes it possible to change its settings.

Drop-down Select

If this check box is selected for a field column, a drop-down icon will appear to the right in the column header. By clicking the icon a list box showing all field values of the field will be opened over the table. You may then make selections and searches in the same manner as if the field had been a row in a multi box.

Label for Column/Row

The text entered here will be shown as title label for the selected dimension or expression, if applicable.

Label for Totals

Here you can specify the text to be shown in the label cells for totals. If no explicit label is specified, the string 'Total' will be used.

Show Partial Sums

Displays partial sums in the pivot table.

Alignment

In this group, the alignment of the expression values and their labels within the pivot table can be set. **Label**, **Data (Numeric)** and **Data (Text)** can be individually set to left-adjusted, centered or right-adjusted. When multi line cells and labels are used, the vertical alignment can be **Top-**, **Center-** or **Bottom-**adjusted.

Allow Pivoting

Deselect this check box to disable the drag-and-drop pivoting.

Vertical Text on Column Labels

If this check box is marked, the column titles are displayed vertically.

Selection Indicators

If this check box is selected selection indicators (beacons) will be shown in table columns containing fields with selections. For new tables this setting takes its default from the **Selection Indicators in Tables** settings on the **Objects** page of **User Preferences**.

Always Fully Expanded

If this check box is selected, all the dimensions are always fully expanded, and the expand/collapse functionality is turned off.

Suppress Expansion Icons in Print

Select this check box if you do not want the '+' and '-' icons for partial expand and collapse to be visible when printing the pivot table.

Suppress Zero Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only null values in all fields in all expressions will

be disregarded in the calculation. This option is selected by default. Turning it off can be useful only in special cases, e.g. if you want to count null values in a chart.

Populate Missing Cells

When this check box is marked, cells in cross tables representing missing combinations of dimensions will be mapped to a regular null value. Thereby it becomes possible to apply expressions testing for null and for attribute expressions and style formats to be applied. This setting is turned on by default for all pivot tables created in QlikView 7.5 and later.

Null Symbol

The symbol entered here will be used for displaying NULL values in the table.

Missing Symbol

The symbol entered here will be used for displaying missing values in the table.

Subtotals

This group contains settings for display of totals and subtotals in the pivot table.

Subtotals on Top

In this option is selected, the totals will be displayed top/left in the pivot table.

Subtotals at Bottom

If this option is selected, the totals will be displayed bottom/right in the pivot table.

Multiline Settings (Expression Data Cells)

In this group you can set the height of the expression data cells.

Wrap Text

If this check box is marked, the contents of the data cells will be wrapped in two or more lines.

Cell Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the cell should have.

Visual Cues

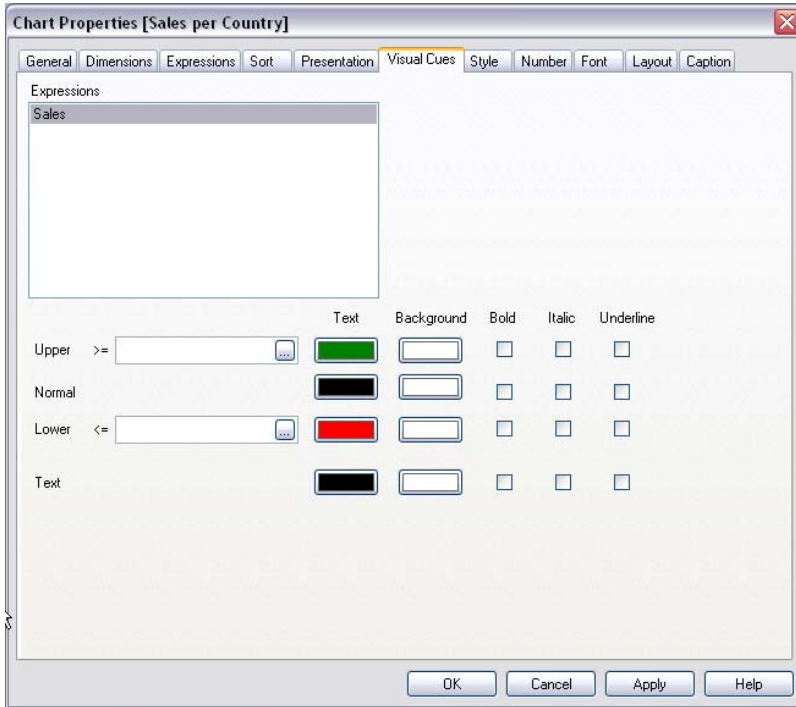


Figure 55. The Visual Cues page for pivot tables in the Chart Properties dialog.

Visual Cues are used to highlight expression values in the table. Values belonging to different value categories can be given separate color and/or font style. Up to four value categories, three different intervals for numeric data plus one for text data, can be individually formatted. Visual cues override table style settings for a given cell.

Expressions

A list of the expressions of the chart. One or more expressions can be selected for formatting of visual cues.

Upper >=

Settings for the upper numeric interval value category. The numeric value above which the upper interval of numeric values begins is stated in the edit box. If this box is left empty no upper interval is defined. The limit may be defined as a calculated formula (see page 467) for dynamic update of its value. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Normal

Settings for the normal numeric interval value category. This interval is defined as all values between the upper limits and lower limits. By default, all numeric values fall within the normal interval.

Lower <=

Settings for the lower numeric interval value category. The numeric value below which the lower interval of numeric values begins is stated in the edit box. If this box is left empty no lower interval is defined. The limit may be defined as a calculated formula (see page 467) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

Text

Settings for text data. This value category is defined as all values without a valid numeric interpretation.

For each value category there are five options making it possible to give the values within the category a specific font, background color, etc.

Text

The color that will be applied to the text of the values. Click the color to redefine it.

Background

The color that will be applied to the cell background. Click the color to redefine it.

Bold

For each value category there is a check box which, if selected, will give values within the category bold font style.

Italic

For each value category there is a check box which, if selected, will give values within the category italic font style.

Underline

For each value category there is a check box which, if selected, will give values within the category underline font style.

Style

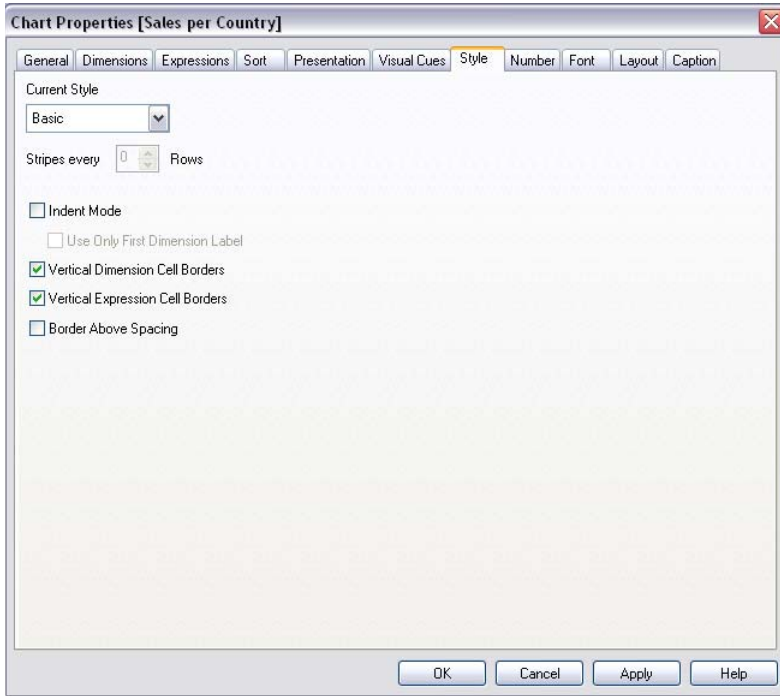


Figure 56. The Style page for pivot tables in the Chart Properties dialog

In the **Style** page, you make settings for the table formatting style.

Current Style

You can choose an appropriate table style from the drop-down list. If the value **[Custom]** appears in the drop-down control a custom style has been applied to the table. If you change the setting back to one of the pre-defined styles, the custom formatting will be lost.

Stripes every n Rows

Here you can see if and at how long intervals shaded stripes appear. Configuration of the control is not possible for pivot tables.

Indent Mode

If this check box is marked, the pivot table's left (column) dimensions will be displayed in indent mode. This means that the dimension values will replace the "Total" label, allowing for a compact display with just small indents between dimensions. The dimension

column area will have a constant width regardless of expand/collapse operations. This setting is only valid for pivot tables.

Use Only First Dimension Label

This setting is only meaningful when a pivot table is in indent mode. By marking this check box the dimension label for the first dimension will occupy the entire label row. Dimension labels for subsequent dimensions will not be shown.

Vertical Dimension Cell Borders

If this check box is deselected all vertical cell borders in dimension cells of the current style will be removed.

Vertical Expression Cell Borders

If this check box is deselected all vertical cell borders in expression cells of the current style will be removed.

Border Above Spacing

If this check box is marked dimension spacing rows (as set in the **Advanced Field Settings** dialog on the **Dimensions** page) will be preceded by the same horizontal borders as those below. Furthermore all vertical borders will be removed within the spacing area.

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the table. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76). The font style cannot be set for pivot tables, since it is defined by the style chosen for the table (see above).

Layout

In the **Layout** page, you can specify how the pivot table should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and

minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83)



13 STRAIGHT TABLE

Sales per CategoryName			
CategoryName	ProductName	Sales	Quantity
		\$1,565,525.31	51952
Men's Clothes	Atlas Lussekoffta	\$30,126.55	1057
Men's Clothes	Bow tie	\$9,534.57	1315
Men's Clothes	Desperado Jeans	\$18,240.68	706
Men's Clothes	Lenin Jeansshorts	\$14,800.64	828
Men's Clothes	M2 Trousers	\$17,944.48	1067
Men's Clothes	O-Man Underwear	\$1,649.87	298
Men's Clothes	Rossi Bermuda Shorts	\$10,947.25	1397
Men's Clothes	Samba Soccer Socks	\$4,941.14	1175
Men's Clothes	US-Master Jeans	\$21,764.94	817
Women's Clothes	Chantell Shirt	\$7,504.70	388
Women's Clothes	Halter Dress	\$361,096.85	981
Women's Clothes	Jack Flash Dress	\$42,638.00	722
Women's Clothes	Langoste Shirt	\$4,433.35	246
Women's Clothes	Le Baby Dress	\$47,571.88	623
Women's Clothes	Minnki Palsi	\$10,472.71	164
Women's Clothes	Okiaba Skin Jackets	\$42,258.78	601
Women's Clothes	Oyaki Kimono	\$9,084.42	806

Figure 57. An example of a straight table

In opposition to the pivot table (see page 201), the straight table cannot display sub-totals or serve as a cross table. On the other hand, any of its columns can be sorted and each of its rows contains one combination of dimension(s)+expression(s).

13.1 Creation

Straight tables are created by choosing **New Sheet Object** from the **Layout** menu, or by pressing the **Create Chart** button in the toolbar. This opens the **Chart Properties** dialog that will help you create the table. Straight tables are logically equivalent to charts.



If you want to create a simple straight table and find the great number of different settings available overwhelming, choose **Create QuickChart** from the list box object menu or from the **Layout** menu (**QuickChart**), or click the **QuickChart** button in the toolbar. The wizard appearing will help you to create a table in a quick and easy way. For more information, see page 265.

13.2 Using the Straight Table

Sorting

It is possible to sort the straight table by any column: simply right-click in the column and choose **Sort** from the context menu. This is equivalent to moving the column to the top of the **Priority** list in the **Sort** page of the straight table **Properties** dialog. An alternative method is to sort by double-clicking the column header.

Rearranging the Columns

You can move the dimension columns and the expression columns by dragging and dropping them with the mouse. Point at the column title, then press the mouse button and keep it depressed while dragging the column to its new position. You may mix dimension and expression columns in any order.

13.3 Straight Table Object Menu

The straight table **Object** menu is found as the **Object** menu when a straight table is active. It can also be opened as a context menu by clicking with the right mouse button on a straight table. The menu contains the following commands:

Properties...

Opens the **Chart Properties** dialog, from which you can set the properties of the table. This command can also be invoked via the keyboard shortcut ALT + ENTER.

Detach

Makes the table static, i.e. cuts the hot link to the selected data.

Attach...

Attaches a detached table, i.e. reestablishes the hot link to the selected data.

Clone

Creates a copy of the table.

Fit Columns to Data

Adjusts the width of all columns in the tables to the widest data in each column. The header is included in the calculation.

Equal Columnwidth

Sets the columnwidth of the other dimension or expression columns of the table equal to that of the column or expression that you have right-clicked on. The command works independently for dimension and expression columns.

Sort

Sorts the table by the column on which you clicked. When the command is used repeatedly on one column, the sort order is reversed each time.

Custom Format Cell

Opens the **Custom Format Cell** dialog (see page 160 in Book II) which lets you format cells in the column you clicked upon. The command is only available when design grid is turned on or when

the **Always Show Design Menu Items** check box is marked (see page 106 in Book I).

Order->

This menu is only available when the design grid is turned on (see page 65 in Book I) or when the **Always Show Design Menu Items** check box is marked (see page 106 in Book I). It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.

Bring to Front

Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.

Send to Back

Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.

Bring Forward

Increases the layout layer of the sheet object by one. Maximum value is 127.

Send Backward

Decreases the layout layer of the sheet object by one. Minimum value is -128.

Clear All Selections

Deselects the values of all the fields in the straight table.

Change Value

Only available for expression columns containing an **inputsum** aggregation of an input field. Sets the cell that has been clicked in input edit mode. Equivalent to clicking the input icon in the cell.

Restore Values

Only available for expression columns containing an **inputsum** aggregation of an input field. Opens a menu with three options.

Restore Single Value

Restores the field values underlying the cell that has been clicked to their default values from the script.

Restore Possible Values

Restores the values of all possible underlying field values to their default values from the script.

Restore All Values

Restores the values of all field values to their default values from the script.

Print...

Opens the standard **Print...** dialog box (see page 263 in Book II), allowing you to print the table.

Print as PDF...

Opens the **Print** dialog with the QlikViewPDF printer preselected. After hitting **Print** you will be prompted for a file name for the PDF output file. This command is only available if a PDF printer is available on the system. The QlikViewPDF printer must be downloaded from the QlikView homepage and installed separately

Send to Excel

Exports the table to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet.

Export...

Exports the contents of the table to a file of your choice. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data files, see page 497 in Book I).

Copy to Clipboard

This menu contains the various copy options for the chart object.

Full Table

Copies the table to the clipboard, complete with header and selection status.

Table Data Area

Copies only the values of the table to the clipboard.

Cell Value

Copies the text value of the list box cell right-clicked upon (when invoking the **Object** menu) to the clipboard.


Image

Copies an image of the chart object to the clipboard. The image will include or exclude the chart object caption and border depending on the settings in the **User Preferences** dialog, **Export** page.


Object

Copies the entire chart object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.


Minimize

Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Maximize

Enlarges the table to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the **Properties** dialog, **Caption** page (see Figure 20 on page 83).

Restore

Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking  in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects.

Help

Opens context specific help.

Remove

Removes the table.

13.4 Chart Properties

When you click the **Create Chart** button in the toolbar, the **Chart Properties** dialog is opened. Here you can set the properties of the chart or table, e.g. the chart type, the dimensions, titles, etc.



Once a table is created, you can change its properties at any time. Choose **Properties** from the chart **Object** menu to open the **Chart Properties** dialog. If the **Properties** command is dimmed, you probably don't have the privileges needed to perform property changes (see page 85 in Book II).

When the desired properties are set, they can be implemented with the **OK** or **Apply** button. The **OK** button also closes the dialog, whereas the **Apply** button does not.

The **Chart Properties** dialog contains a number of pages, which are described below:

General

On the **General** page (Figure 1 on page 21), you set the chart type, choose a name for the chart or table etc. This page is the same for all the chart types.

Dimensions

On the **Dimensions** page, the dimensions to be shown in the table can be set. This page is identical with the **Dimensions** page of the bar chart (Figure 4 on page 31).

Expressions

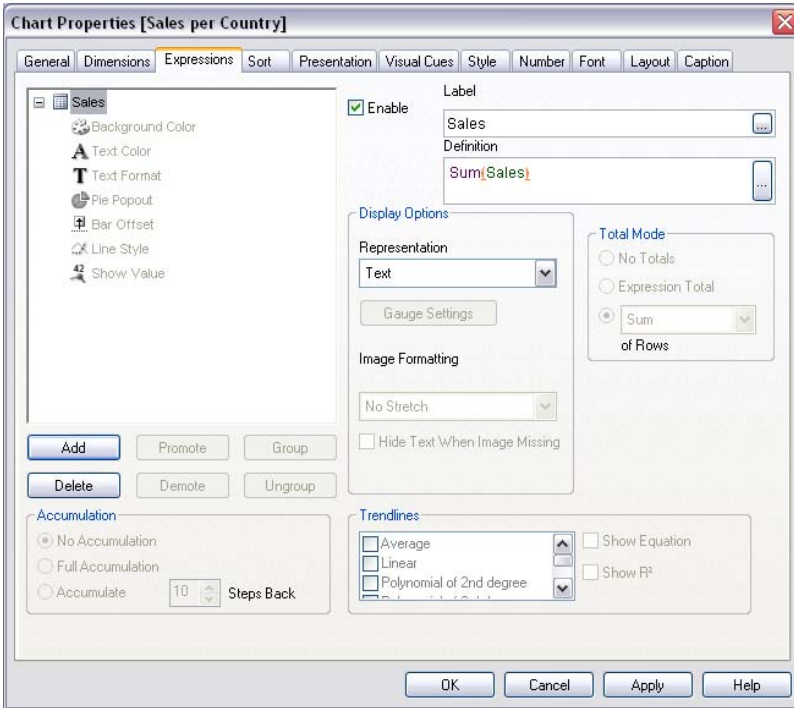




Figure 58. The Expressions page for straight tables in the Chart Properties dialog

On the **Expressions** page you set the expressions to be displayed in the table. The Expressions page for table charts differ in a number of ways to the corresponding page for bitmap charts. As an example trendlines and accumulation expressions are not available in straight tables.


Expressions

The upper left corner of the dialog contains a list of the chart's expressions. An expression defines the calculated contents of a chart, e.g. the data shown in an expression column in the table.

Navigating the Tree Control

The expression list is in fact a tree control with a large set of control options. In front of each expression you will find an expansion icon . By clicking the expansion icon you open up underlying sub expressions or attribute expressions. The icon will then shift to a collapse icon . By clicking that icon you collapse the display of any sub expressions or attribute expressions. In complex cases up to three levels of expansion may be possible.


Display Options / Plot Modes

After the expansion/collapse icon on each row you will see an icon corresponding to the type of display to be used for the expression. These are normally set in the **Display Options** group (see below). There is for example one icon **T** for expressions plotted as plain text, one  for expressions plotted as gauges in cells and so on. Only icons reflecting plot types possible in the active chart type will be shown.

Sub expressions

Certain plot options in bitmap charts make use of sub expressions, i.e. a set of two or more expressions that together define the plot symbol. If **Stock** or **Box** plot has been selected as **Display Option** for an expression in a combo chart, that expression will be displayed as an empty placeholder with four or five sub expressions when changing the chart type into a table chart. The display options in the table chart can be set individually for each of the sub expressions. All expression definitions lie within the sub expressions.

Attribute expressions

Expression data can be dynamically formatted by means of attribute expressions. Click on the  expansion icon in front of any expression to display the placeholders for the dimension's attribute expressions. You may choose to use any number and combination of attribute expressions on a given base expression. Formatting defined by means of attribute expression supersede all formatting implied by table style or visual cues. Whenever an attribute expression is entered for a dimension, its icon will turn from gray scale to color, or as in the case of **Text Format** from gray to black. The available attribute expression types are:

 **Background Color**

Click on the **Background Color** in order to define an attribute expression for calculating the table cell background

color of the data point. The calculated color will have precedence over table style and visual cues. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, white background will be used.


Text Color

Click **Text Color** in order to define an attribute expression for calculating the table cell text color of the data point. The calculated color will have precedence over table style and visual cues. The expression used should return a valid color representation (a number representing the Red, Green and Blue components as defined in Visual Basic), which is typically achieved by using one of the special chart color functions (see page 443). If the result of the expression is not a valid color representation, black will be used.

T Text Format

Click **Text Format** in order to enter an attribute expression for calculating the font style of text in the table cell for each dimension cell. The calculated text format will have precedence over table style. The expression used as text format expression should return a string containing a B for bold text, I for italic text and/or U for underlined text.

Expression Cycle Groups

Finally there is the concept of expression groups. Any number of base expression (with their respective sub expressions and attribute expressions) may be bundled together in a group. A group is shown in the list as a placeholder with a  cycle icon. By expanding the group, you can see the expressions it contains. In the chart only the first expression in the list will be used for display. The expression to be shown out of the group can be changed by clicking the cycle icon in the chart or in the table expression column header. By right-clicking the cycle icon, you get a pop-up list of the available expressions for direct selection. When you cycle the group the second expression will be promoted to the top of the list and the previously active expression demoted to the bottom of the list.

You can add an expression to a group in two ways. The first method is by simply dragging an expression on top of another expression in

the list (creates a new cycle group) or on top of an existing cycle group in the list (adds the expression to the existing group). The second method is by selecting an expression in the list and then click on the **Group** button (see below). The expression will then be grouped with the expression or cycle group residing above it in the list.

You can correspondingly remove an expression from a group by dragging it out of the group in the list or by selecting it and pressing the **Ungroup** button (see below). If only one expression remains in the cycle group after the extraction, that expression will also be extracted and the cycle group will cease to exist.

Add

By clicking this button, you get to the **Edit Expression** dialog (page 249), where it is possible to create new expressions.

Delete

Deletes the selected expression.

Promote

Promotes the selected expression one step up in the list. Use this button if you want to change the order of the expressions.

Demote

Demotes the selected expression one step down in the list. Use this button if you want to change the order of the expressions.

Group

It is possible to group two or more expressions, so that only one of them is displayed in the chart at any given time. The expression to be shown out of the group can be changed by clicking the cycle icon in the chart or in the table expression column header. By right-clicking the cycle icon, you get a pop-up list of the available expressions for direct selection. You may assign an expression to a cycle group by selecting an expression in the list and then click on the **Group** button. The expression will then be grouped with the expression or cycle group residing above it in the list.

Ungroup

This command is only available when an expression belonging to a cycle group has been selected in the Expressions list above. By clicking the button the expression will be moved out of the cycle group and placed as an independent expression after the cycle group in the list. If only one expression remains in the cycle group after the extraction, that expression will also be extracted and the cycle group will cease to exist.

Label

The label of the expression. Enter the name you wish to display in the chart. If no text is entered, the label will be automatically set to the expression text. The text may be entered as a calculated expression for automatic update.

Definition

Shows the composition of the selected expression. It is possible to edit the expression directly in this box. By clicking the ... button the full **Edit Expression** dialog is opened.

Enable

Marking this check box will enable the selected expression. If the box is not marked, the expression will not be used at all.

Relative

Marking this check box will set the chart to show the result in percent instead of absolute numbers. This option is not available for straight tables.

Total Mode

This group is not available for straight tables. By selecting one of the options in this group, you decide how the total of the selected expression is to be calculated. This setting is important for relative display or when showing totals.

No Totals

If this option is selected no total will be calculated for the expression.

Expression Total

If this option is selected, the expression total will be calculated using all the values of the field. If, e.g., the selected column contains the average salaries for different business categories, choosing the Expression Total option will result in the average salary for all the business categories.

F(x) of Rows

If this option is selected, the individual values of each data point (each bar in a bar chart, each row in a straight table etc.) for the selected expression will be summed up aggregated using the selected aggregation function (typically summed up).

Display Options

This group defines what will be written in the table chart expression cells.

Representation

The following alternatives are given:

Text

When selecting this option the expression values will always be interpreted and displayed as text.

Image

When selecting this option QlikView will try to interpret each expression value as a reference to an image. The reference may be a path to an image file on disk (e.g. `C:\MyPic.jpg`) or inside the qvw document (e.g. `qmem://<Name>/<Peter>`). If QlikView cannot interpret an expression value as a valid image reference, the value itself will be displayed.

Circular Gauge

When selecting this option QlikView will display the expression value in a circular style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

Linear Gauge

When selecting this option QlikView will display the expression value in a horizontal linear style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

Traffic Light Gauge

When selecting this option QlikView will display the expression value in a horizontal traffic light style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

LED Gauge

When selecting this option QlikView will display the expression value in a LED style gauge. The gauge chart will be inscribed in the available table cell. The visual settings for the gauge can be modified via the **Gauge Settings...** button.

Link

Select this option to enter an expression in the **Definition** field that will create a clickable link in the table cell. The expression should return a text that can be interpreted as *DisplayText*<*url*>*LinkText*. The *DisplayText* will be displayed in the table cell and *LinkText* will be the link that is opened in a new browser window. If a link is defined, the value in the table cell will be underlined. If no link is defined the value will not be underlined. Note that it is not possible to make selections in a cell with **Link** as display mode. By clicking the ... button the full **Edit Expression** dialog is opened.

Examples:

```
=Name & '<url>' & Link
```

```
=Name & '<url>www.qlikview.com'
```

where *Name* and *Link* are table fields loaded in the script.

Mini Chart

When selecting this option QlikView will display the expression value in a miniature chart with the expression aggregated over an extra dimension. The chart will be inscribed in the available table cell. The mini chart dimension can be defined and visual settings for the chart can be modified via the **Mini Chart Settings** button (see below).

Hide

When selecting this option the expression is not shown in the table.

Gauge Settings...

Only available when one of the gauge options has been selected above. Opens a dialog for setting the properties of the gauge. This dialog is essentially equal to the **Presentation** page of Chart Properties for gauge charts (see page 194).

Mini Chart Settings

Only available when **Mini Chart** has been selected as **Display Option** for an expression in the table. Opens a dialog for setting the properties of the mini chart.

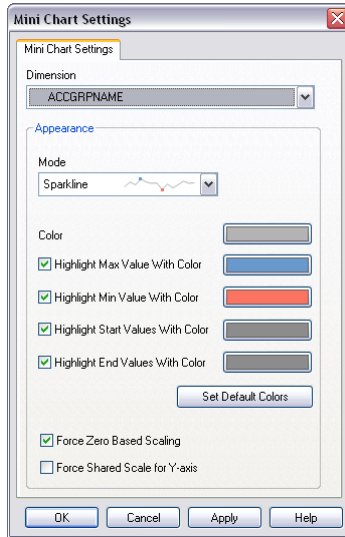


Figure 59. Mini Chart settings dialog

Dimension

Choose the dimension for which the expression should be plotted.

Mode

Set the mini chart as sparklines, lines, dots, bars, or whiskers.

Color

Opens the **Color Area** dialog where the plot color of the mini chart can be set.

Highlight Max Value with Color

Mark the check box and click on the colored button to open the **Color Area** dialog where the color for the maximum value can be set.

Highlight Min Value with Color

Mark the check box and click on the colored button to open the **Color Area** dialog where the color for the minimum value can be set. The setting is only available when the **Mode** is set to **Sparkline**.

Highlight Start Values with Color

Mark the check box and click on the colored button to open the **Color Area** dialog where the color for the start values can be set. Not available for **Bars** or **Whiskers**.

Highlight End Values with Color

Mark the check box and click on the colored button to open the **Color Area** dialog where the color for the end values can be set. Not available for **Bars** or **Whiskers**.

Set Default Colors

Sets the colors for the highlight settings to QlikView default colors.

Force Zero Based Scaling

Fixes the lower edge of the chart to zero on the axis. Not available for whiskers.

Force Shared Scale for Y-Axis

Forces all cells in the column to use the same y-axis scale.

Image Formatting

Only available when the image options have been selected above. This setting describes how QlikView formats the image to fit in the cell. There are four alternatives.

No Stretch

If this option is selected, the image will be shown as is, without any stretching. This may cause parts of the picture to be invisible or only part of the cell to be filled.

Fill

If this option is selected, the image will be stretched to fit the cell without bothering about keeping the aspect ratio of the image.

Keep Aspect

If this option is selected, the image will be stretched as far as possible to fill the cell while keeping the aspect ratio. This typically results in areas either on both sides of or above and below which are not filled by the image.

Fill with Aspect

If this option is selected, the image will be stretched to fill the cell in both directions while

keeping the aspect ratio. This typically results in cropping of the image in one direction.

Hide Text When Image Missing

If this option is selected, QlikView will not display the field value text if interpretation as an image reference fails for some reason. The cell will then be left blank.

Sort

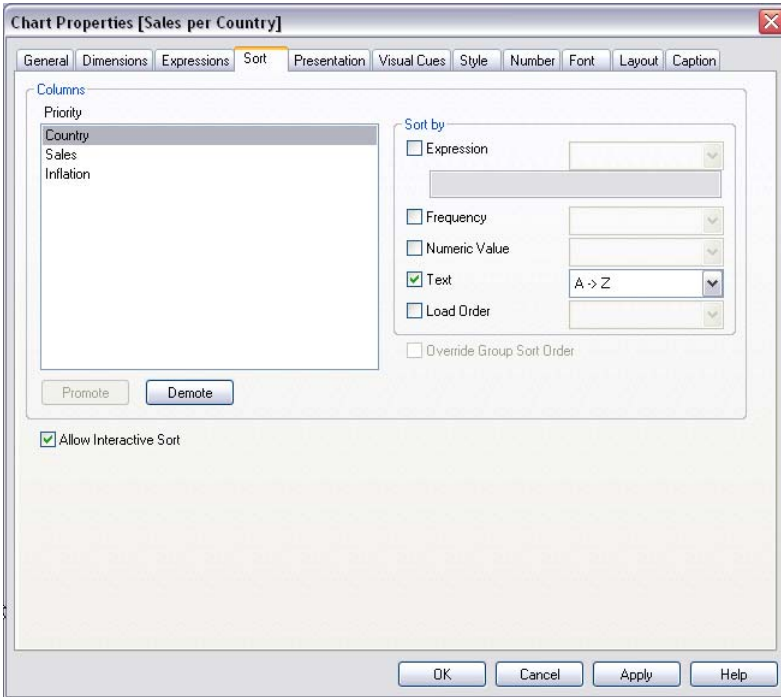


Figure 60. The Sort page for straight tables in the Chart Properties dialog

In this page you set the sort order of the columns as well as that of the column values.

Columns

In the **Columns** group you set sort options for the individual table columns. The table is sorted according to the sort criteria set for the column at the top.

Priority

Contains the dimensions/expressions chosen as columns listed in sort priority. Selecting one of the columns makes it

possible to set the sort order of its values in the **Sort by** group, as well as to change the sort priority by clicking one of the buttons **Promote** or **Demote**.

Promote

Moves the selected column one step further up in the **Priority** box above.

Demote

Moves the column selected one step further down in the **Priority** box above.

Sort by

In this group, the sort order of the column values can be set.

Expression

Sorts the column values according to the expression entered into the text edit box below this sort option. This option is not available for calculated dimensions.

Frequency

Sorts the column values by frequency (number of occurrences in the table).

Numeric Value

Sorts the column values by their numeric values.

Text

Sorts the column values in alphabetical order.

Load Order

Sorts the column values by the initial load order.

Override Group Sort Order

This check box is only available when a group dimension is selected in the Priority list. Normally the sort order of a group dimension is determined for each field in a group via the group properties. By marking this check box you can override any such settings on group level and apply a single sort order for the dimension, regardless of which field is active in the group.

The order of priority is **Expression, Frequency, Numeric Value, Text** and **Load Order**. Each of these sort criteria can be set to ascending or descending.

Allow Interactive Sort

Deselecting this check box will disable the **Sort** command of the object menu.

Presentation

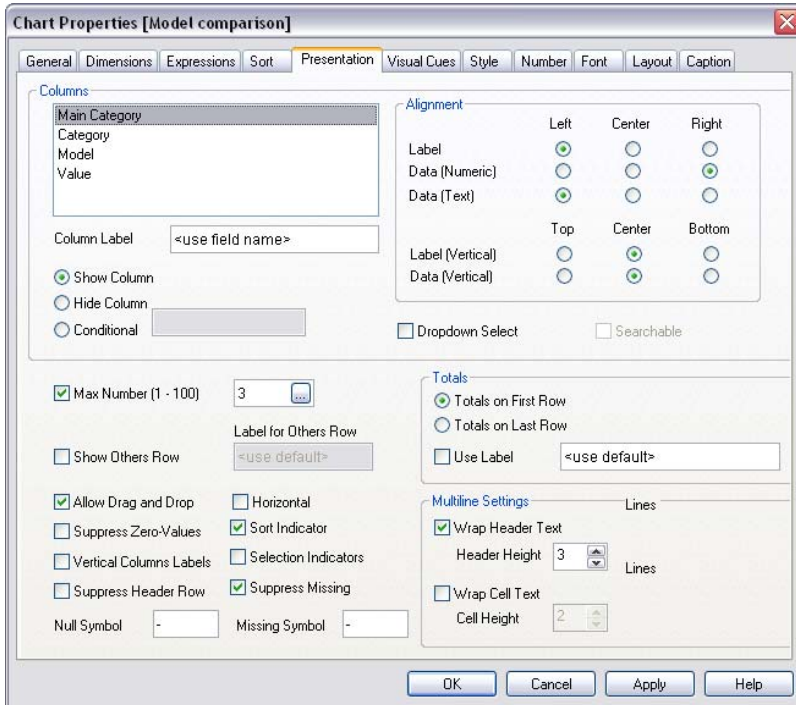


Figure 61. The Presentation page for straight tables in the Chart Properties dialog

By changing the settings on this page, you modify the way the straight table is presented on the screen.

Columns

A list of the dimensions and expressions chosen as columns. Selecting one of the columns makes it possible to change its settings.

Show Column

By selecting this radio button, the selected column will be visible when the table is drawn in the layout.

Hide Column

By selecting this radio button, the selected column will be hidden when the table is drawn in the layout. This may be useful when you want to sort the table by a column which you do not want to show to the application user.


Conditional

The column will be shown or hidden depending on a conditional expression which will be evaluated each time the table is drawn. The column will only be visible when the condition returns true. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas (see page 249).

Drop-down Select

If this check box is selected for a field column, an icon will appear to the right in the column header. By clicking the icon a list box showing all field values of the field will be opened over the table. You may then make selections and searches in the same manner as if the field had been a row in a multi box.

Searchable

If this check box is selected for an expression column, a  search icon will appear to the left in the column header. By clicking the icon a search box will be opened. You may then type a search criteria (e.g. >100000). When hitting ENTER all table rows with an expression value matching the search criteria will be selected.

Column Label

The text entered here will be shown on the title bar of the specific column.

Alignment

In this group, the alignment of the column values and their labels can be set. **Label**, **Data (Numeric)** and **Data (Text)** can be individually **Left**-, **Center**-, or **Right**-adjusted. When multi line cells and labels are used, the vertical alignment can be **Top**-, **Center**- or **Bottom**-adjusted.

Max Number

Here you can set the maximum number of rows to display. Rows displaying sums are not counted; however, sums will still be calculated using all the rows of the table.

Show Others Row

This option is available only if the number of displayed rows has been limited (Max Number). Mark this check box to display a row for the surplus values. The **Others** row is calculated as a simple sum of the expression values of the rows omitted by the Max Number

Setting. The **Others** bar will thus only be meaningful for expressions where the **Sum of Rows** total is meaningful.

Label for Others Row

Here you can specify a label for the Others row. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default Others label found in the **Objects** page of the **User Preferences** dialog will be used (see page 107 in Book I).

Horizontal

Mark this check box to display the straight table transposed 90 degrees, so that labels from a left column and each table record becomes a column of data.

Vertical Column Labels

If this check box is marked, the column headers as vertical text.

Suppress Header Row

If this check box is marked the table will be shown without header (label) row.

Allow Drag and Drop

Deselecting this check box will disable drag and drop column moves within the table.

Sort Indicator

If this check box is selected, a sort indicator icon (arrow) will be shown in the header of the column, which is at the top of the current column sort order. The direction of the icon indicates whether the column is sorted ascending or descending.

Selection Indicators

If this check box is selected selection indicators (beacons) will be shown in table columns containing fields with selections. For new tables this setting takes its default from the **Selection Indicators in Tables** setting on the **Objects** page of **User Preferences**.

Suppress Zero Values

If this check box is marked, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is selected by default.

Suppress Missing

If this check box is marked, all combinations of the dimension fields associated with only NULL values in all fields in all expressions will be disregarded in the calculation. This option is selected by

default. Turning it off can be useful only in special cases, e.g. if you want to count NULL values in a chart.

Null Symbol

The symbol entered here will be used for displaying NULL values in the table.

Missing Symbol

The symbol entered here will be used for displaying missing values in the table.

Totals

This group contains settings for total display.

Totals on First Row

If this option is selected, the totals will be displayed on the first row of the table.

Totals on Last Row

If this option is selected, the totals will be displayed on the last row of the table.

Use Label

Here you can specify a label for the totals. The label may be defined as a calculated formula (see page 467) for dynamic update. If nothing is specified, the default total label found in the **Objects** page of the **User Preferences** dialog (see page 107 in Book I).

Multiline Settings

In this group you can set the height of the table header and data row cells.

Wrap Header Text

If this check box is marked, the contents of the header row cells will be wrapped in two or more lines.

Header Height (Lines)

If you have selected **Wrap Header Text**, this is where you set how many lines the header cells should have.

Wrap Text

If this check box is marked, the contents of the data cells will be wrapped in two or more lines.

Cell Height (Lines)

If you have selected **Wrap Text**, this is where you set how many lines the cell should have.

Visual Cues

Visual Cues are used to highlight expression values in the table. Values belonging to different value categories can be given separate color and/or font style. This page is identical with the **Visual Cues** page of the pivot table (Figure 55 on page 221).

Style

In the **Style** page, you can choose an appropriate table style from the drop-down list. In the edit box below, it is possible to set shaded stripes for straight tables and to specify at how long intervals these should appear. This page is identical with the **Style** page of the pivot table (Figure 56 on page 223).

Number

In the **Number** page, the display number format can be set. This page is identical with the **Number** page of the bar chart (Figure 16 on page 75).

Font

In the **Font** page, you can change the font of the text in the table. This page is identical with the **Font** page of the bar chart (Figure 17 on page 76). The font style cannot be set for straight tables, since it is defined by the style chosen for the table (see above).

Layout

In the **Layout** page, you can specify how the straight table should appear on the layout. This includes settings for shape, border and the layer on which the chart should reside. This page is identical with the **Layout** page of the bar chart (Figure 18 on page 77).

Caption

In the **Caption** page advanced settings can be made for the caption. Background and foreground (text) color can be defined with separate settings for active and inactive state. Furthermore there are settings for multi-line text and text alignment in both vertical and horizontal direction. Maximize and minimize options are also found on this page. This page is identical with the **Caption** page of the bar chart (Figure 20 on page 83).



14 EDIT EXPRESSION

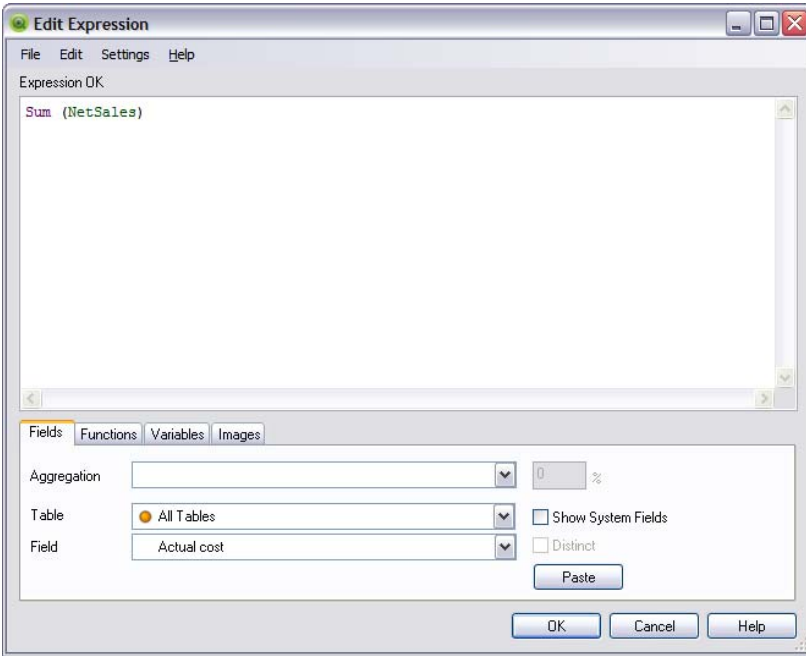


Figure 62. The Edit Expression dialog

In this dialog it is possible to define an expression for use in the QlikView layout (in charts and elsewhere), using all the operators and functions described in “Chart Expressions” on page 293 and forward. The dialog is fully sizeable in order to facilitate editing of large and complex expressions.

Expression

The current expression, consisting of one or several statistical functions, operators and constants. You can type an expression directly into this box or compose an expression using the buttons and boxes found in the tabs below. The expression will be color coded for syntax control.

The text above the edit control will change to reflect the correctness of the expression currently displayed. When a syntactically correct expression is present, the text will read **Expression OK**. If the expression contains syntax errors an error message will be shown. It

is possible to leave the dialog with a syntactically incorrect expression, but the result will be as might be expected.

OK

Accepts changes made in the expression and closes the dialog box. The changes will be saved next time the document is saved to a file.

Cancel

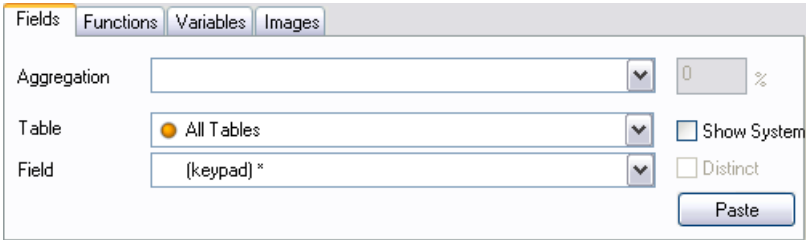
Cancels the expression editing and closes the dialog box. The expression will be restored.

Help

Opens the **Help** dialog for expressions.

In the bottom pane of the dialog you will find a number of tabs for various tasks in relation to the definition of QlikView expressions.

Fields



The screenshot shows a dialog box with four tabs: 'Fields', 'Functions', 'Variables', and 'Images'. The 'Fields' tab is active. It contains three rows of controls: 'Aggregation' with a dropdown menu and a percentage symbol; 'Table' with a dropdown menu showing 'All Tables' and a 'Show System' checkbox; and 'Field' with a dropdown menu showing '(keypad) *' and a 'Distinct' checkbox. A 'Paste' button is located at the bottom right.

In the **Fields** tab you will find controls for pasting syntax relating to QlikView field data.

Aggregation

In this drop-down you can choose from the statistical aggregation functions available in the QlikView layout.

Table

In this drop-down you may select a specific input table from which to pick fields in order to make navigation in the **Field** drop-down easier.

Field

This drop-down lists all the fields available. The list may be narrowed down by selecting a specific input table in the **Table** drop-down above.

Show System Fields

If this check box is marked, the list containing the fields of the document includes the system fields.

Distinct

The statistical functions are by default calculated on the number of occurrences in the original table. Sometimes, however, you do not want to calculate duplicates. If this is the case, mark this check box before pasting the function.

Paste

Pastes the selected function or just the field into the **Expression** box.

Functions

The screenshot shows the 'Functions' tab in the QlikView interface. At the top, there are four tabs: 'Fields', 'Functions' (which is highlighted), 'Variables', and 'Images'. Below the tabs, there are two dropdown menus. The first is labeled 'Function Category' and is set to '--- All Functions ---'. The second is labeled 'Function Name' and is set to 'acos'. To the right of the 'Function Name' dropdown is a 'Paste' button. Below these elements is a text box containing the text 'num acos (x)'. The text box has a light gray background and a thin border.

In the **Functions** tab you will find controls for pasting syntax relating to QlikView general functions.

Function Category

In the drop-down you may select a category of functions in order to make navigation in the **Function Name** drop-down easier.

Function name

In the drop-down you may select a function for pasting into the expression out of all functions available in the QlikView layout. The list can be reduced to show only functions belonging to a certain category by means of a selection in the **Function Category** drop-down above.

Paste

Pastes the selected function name into the **Expression** box.

At the bottom of the tab there is a pane showing the argument syntax of the function selected in the **Function Name** drop-down.

Variables



In the **Variables** tab you will find controls for pasting syntax relating to QlikView variables.

Variables

In the drop-down you will find all currently defined variables in the document.

Paste

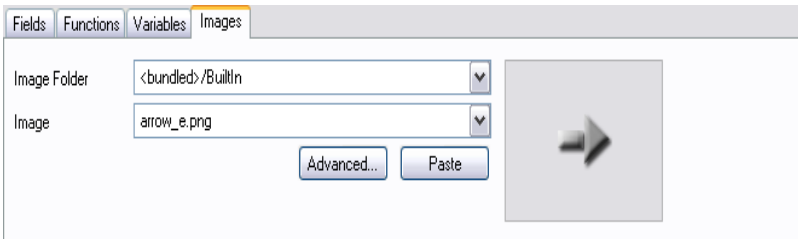
Pastes the selected function into the **Expression** box.

Show System Variables

If this check box is marked, the list in the **Variables** drop-down will include the system variables.

At the bottom of the tab there is a pane showing the current value of any variable selected in the **Variables** drop-down.

Images



In the **Images** tab you will find controls for pasting syntax for accessing the pictures built into QlikView (see page 205 in book I). Note that this functionality is only relevant in certain parts of the QlikView layout,

Image Folder

In this drop-down you select the image folder.

Image

In this drop-down you select the image.

Advanced...

Opens up an image browser dialog showing all the available images.

Paste

Pastes the syntax for accessing the selected built-in image into the **Expression** box.

To the right on the tab there is a pane showing a preview of the image selected in the **Image** drop-down.

FILE menu**Export to Expression File...**

Saves the contents of the **Expression** box in a text file to be specified in the **Save Script** dialog that appears. The file will have the extension .qve.

Insert File...

Lets you browse for a file containing an expression or part of an expression and inserts its contents into the **Expression** box at the position of the cursor.

Colormix Wizard...

Opens the **Colormix Wizard** (see page 254) that facilitates the creation of color mix expressions, i.e. expressions that calculate a color based on a given measure.

Print...

Opens the Windows standard **Print** dialog for printing the expression. This command can also be invoked via the keyboard shortcut CTRL+P.

EDIT menu**Undo**

Undoes the latest change. This command can also be invoked via the keyboard shortcut CTRL+Z.

Redo

Redoes the latest **Undo**. This command can also be invoked via the keyboard shortcut CTRL+Y.

Cut

Exports the selected text to the clipboard. This command can also be invoked via the keyboard shortcut CTRL+X.

Copy

Copies the selected text to the clipboard. This command can also be invoked via the keyboard shortcut CTRL+C.

Paste

Pastes the contents of the clipboard into the dialog at the position of the cursor. This command can also be invoked via the keyboard shortcut CTRL+V.

Clear

Clears the entire expression.

Select All

Selects the entire expression.

Find/Replace...

Opens a dialog allowing you to find and replace numbers or characters in the expression. This command can also be invoked via the keyboard shortcut CTRL+F.

SETTINGS menu

Configure...

Opens the **Editor** page in the **User Preferences** dialog (page 99 in Book I), where you can set the font and color of the different text types appearing in the **Expression** box.

14.1 The Colormix Wizard

With the **Colormix Wizard**, opened from the **File** menu in the **Edit Expression** dialog, it is possible to create a color mix expression, i.e. an expression that calculates a dynamic color from a given measure.

It could e.g. be a calculated index with a value around 100 displayed in a pivot table. The higher index, the better. If a specific dimension value had an index that vastly exceeded 100, one would probably want to mark this pivot table line green, whereas red probably would be the appropriate color if the index value was much lower than 100. And one would want the color change to be gradual with yellow denoting 100. In other words – you want a dynamic color.

Such a gradual color calculation can be done with the **Colormix** functions in QlikView, but it can be cumbersome to create the appropriate expression inside the Colormix function. Here, the **Colormix Wizard** can help.

First, it is important to clarify that the color functions, hence also the **Colormix Wizard**, are only relevant to use in place where QlikView expects a color function, i.e. not in the chart expression itself, but rather in the **Background Color** expression or in the **Text Color** expression.

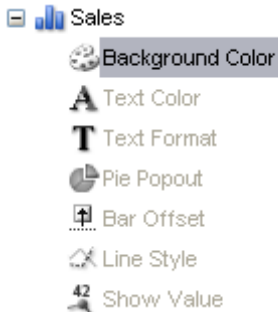


Figure 63. The Background Color expression

When you open the **Colormix Wizard** from the **File** menu in the **Edit Expression** dialog, you will first see a page describing what is needed to create a dynamic color. If you want to skip the start page when you use the wizard in the future, mark the **Don't show this page again** check box.

Step One – Enter a value expression

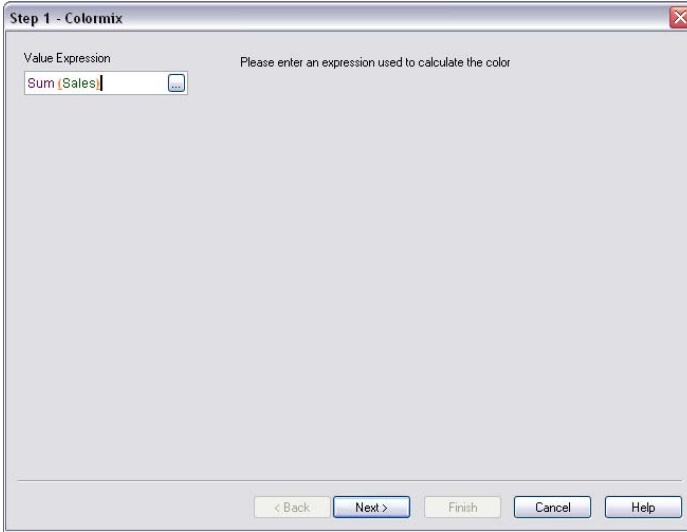


Figure 64. Step one of the Colormix Wizard

This is the expression that will control what color QlikView will show. Typical expressions could be

- $\text{Sum(Sales)} / \text{Sum(total Sales)}$
- $\text{Sum(Sales)} / \text{Sum(Quota)}$
- Avg(Age)

Note that it is usually a value that has the same order of magnitude no matter how many or how few records you have selected. An average, a percentage or an index is usually a good measure to use.

Step Two – Set upper and lower limits

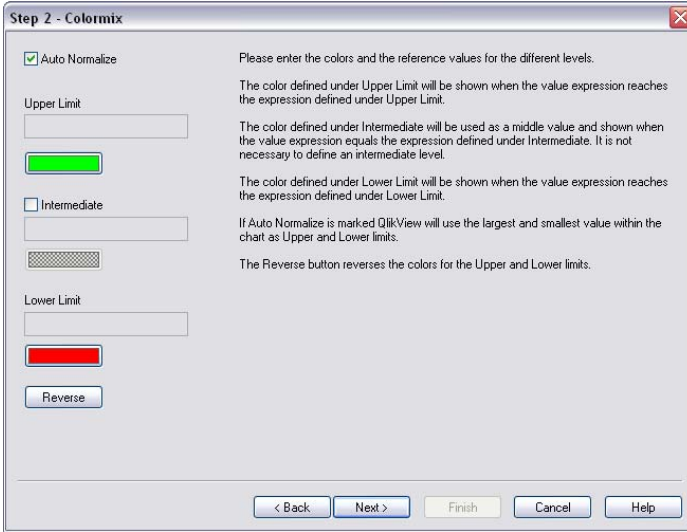


Figure 65. Step two of the Colormix Wizard

In this page you need to define the upper and lower limits of the value expressions and the corresponding colors. “Limit” meaning not the maximum possible value, but the limit where the maximum color is reached.

Typical expressions could e.g. be

Value Expression	Upper limit	Lower limit
$\text{Sum}(\text{Sales}) / \text{Sum}(\text{total Sales})$	$\text{Sum}(\text{total Sales})$	0
$\text{Sum}(\text{Sales}) / \text{Sum}(\text{Quota})$	2 (=200%)	0
$\text{Avg}(\text{Age})$	$\text{Max}(\text{total Age})$	$\text{Min}(\text{total Age})$

But other limits are possible of course.

If you use the color wizard within a chart, then the following two expressions are always a good choice:

- **RangeMax (top(total <ValueExpression>,1,NoOfRows(total)))**
- **RangeMin (top(total <ValueExpression>,1,NoOfRows(total)))**

These expressions will calculate the largest and the smallest row value of <ValueExpression> within the chart.

Auto Normalize

If this check box is marked, QlikView will try to find proper upper and lower limits. In such a case, expressions for the upper and lower levels cannot be entered manually.

Upper limit

Here an expression for an upper limit must be entered, unless the **Auto Normalize** option is checked.

Intermediate

Here it is possible to enter an expression for an intermediate level, linked to a third color.

Lower limit

Here an expression for a lower limit must be entered, unless the **Auto Normalize** option is checked.

Reverse

This button reverses the colors for the upper and lower limits.

Click **Next** to continue.

Step Three – Finalize

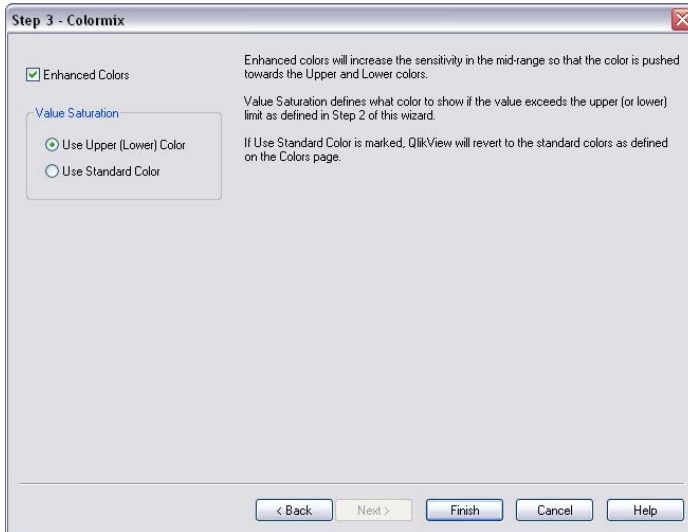


Figure 66. Step three of the Colormix Wizard

In this page you finalize the color mix expression

Enhanced colors

If this check box is marked, a hysteresis transformation is applied on top of the value expression. This will increase the sensitivity in the mid-range so that the color is pushed towards the **Upper** and **Lower** colors.

Value Saturation

Here it is possible to control the behavior if the value expression exceeds the upper or lower limit. If **Use Upper (Lower) color** is marked, QlikView will use the color for the maximum or the minimum. If **Use Standard color** is marked, QlikView will instead revert to the standard colors as defined on the **Colors** page.



15 THE BOX PLOT WIZARD

The Box Plot Wizard creates a combo chart, often used for display of statistical data. The expression will be preceded by the icon in the Expressions list and appear as an empty placeholder with five sub expressions. Read more about the box plot on page 46.

15.1 Starting the Box Plot Wizard

You start the wizard from the **Tools** menu.

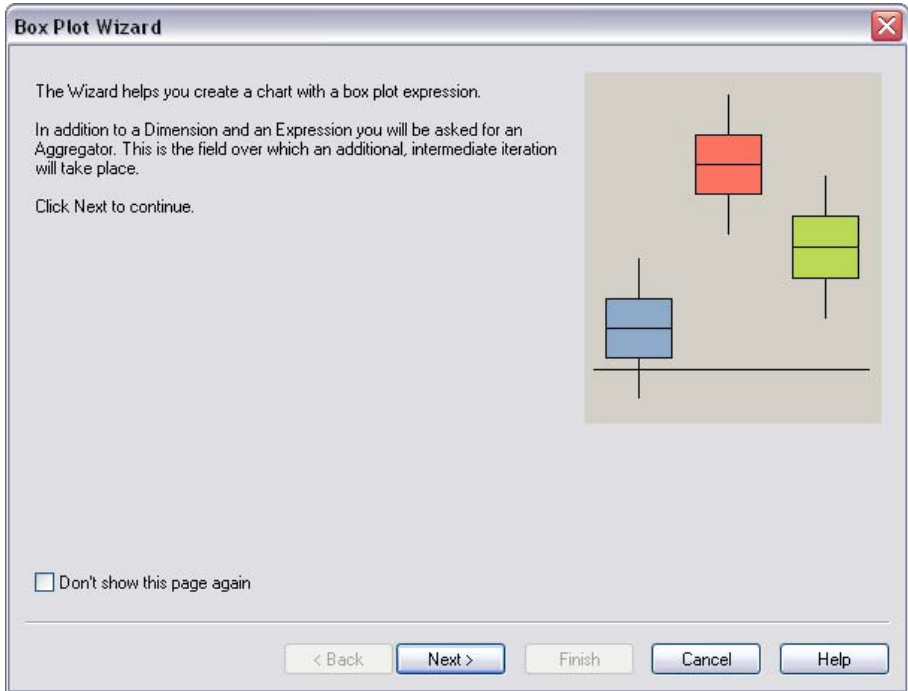


Figure 67. The start page of the box plot wizard

The first time you start the wizard you will be met by a start page outlining the purpose of the wizard and the basic steps involved. If you want to skip the start page when you use the wizard in the future, mark the **Don't show this page again** check box.

15.2 Pages of the Box Plot Wizard

Step 1- Define data

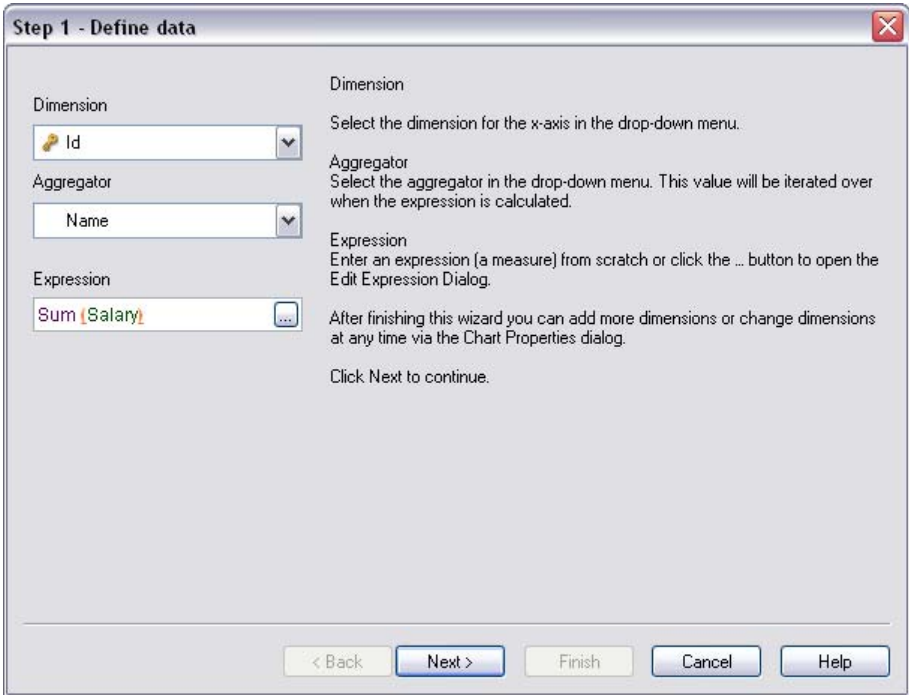


Figure 68. Step 1 of the box plot wizard

Select the **dimension** for the x-axis. After finishing this wizard you can add more dimensions or change dimensions at any time via the **Chart Properties** dialog.

Select the **aggregator**. The aggregator is the value that is iterated over when the expression is calculated.

Lastly, define the **expression**. The expression defines the calculated value in the chart. They are typically found on the y-axis in a bar chart. Expressions in QlikView can range from short and simple to long and complex. This field allows you to type your expression from scratch.

After finishing this wizard you can change the expression or add more expressions at any time via the **Chart Properties** dialog.

Step 2 Presentation



Figure 69. Step 2 of the box plot wizard

Display Mode

Choose the **Display Mode** for the distribution, **Average** or **Median**.

Include Whiskers

Include whiskers to have the chart display the upper and lower whiskers, as either **Min/Max** or **5/95 percentiles**.

Use Outliers (Min/Max)

Mark this check box to show so called outliers for extreme values.



16 THE QUICK CHART WIZARD

The **Quick Chart** feature has been developed for the user who wants to create a simple chart in a quick and easy way, without bothering about the great number of different settings and options available. The result is however always a full-blown chart - it is only the creation process that has been simplified.

When you click the **Quick Chart Wizard** button in the toolbar (or choose **Create Quick Chart** from the list box object menu, or **Quick Chart Wizard** from the **Tools** menu), the **Quick Chart** wizard appears.

The quick chart wizard takes you through the following basic steps:

- 1 Select chart type
- 2 Define chart dimension(s)
- 3 Define the chart expression
- 4 Format chart

Depending on selections some of the above steps may be skipped.

16.1 Starting the Quick Chart Wizard

You start the quick chart wizard either by choosing **Quick Chart Wizard** from the **Layout** menu or by clicking the button in the **Design** toolbar.



Figure 70. The start page of the quick chart wizard

The first time you start the wizard you will be met by a start page outlining the purpose of the wizard and the basic steps involved. If you want to skip the start page when you use the wizard in the future, mark the **Don't show this page again** check box.

Click **Next** to continue.

16.2 Pages of the Quick Chart Wizard

Step 1 - Select Chart Type

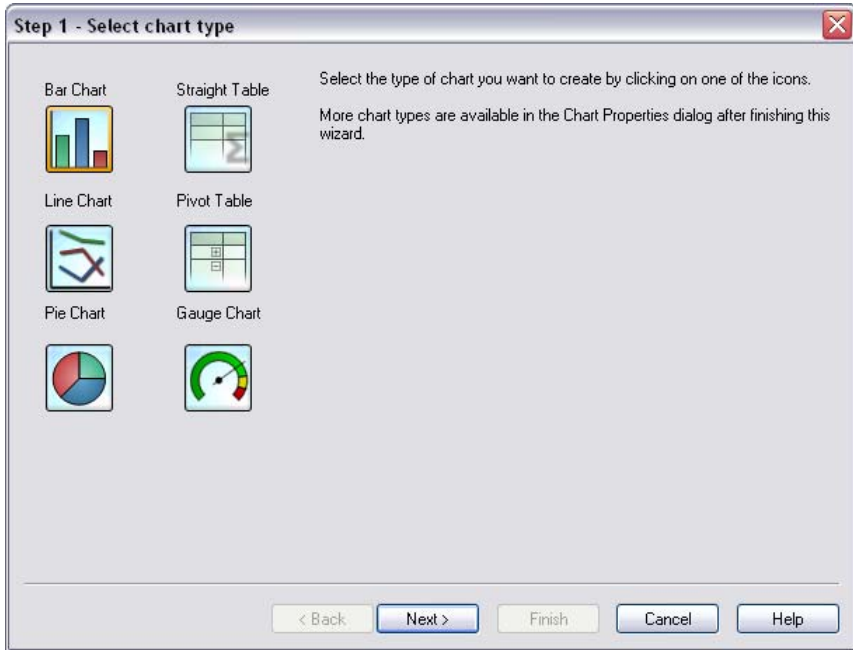


Figure 71. The first page of the quick chart wizard

Select the type of chart you want to create by clicking on one of the icons.

The chart types available are those most commonly used in QlikView. You may change the chart into any other QlikView chart type via the **Chart Properties** dialog after finishing the wizard.

Click **Next** to continue.

Step 2 - Define Dimension(s)

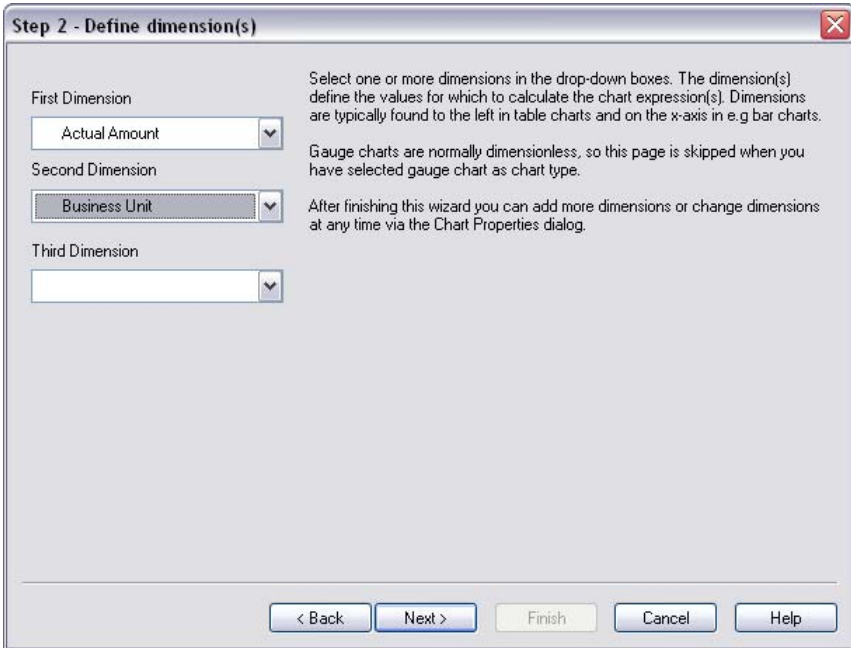


Figure 72. The second page of the quick chart wizard

Unless you selected Gauge chart in the first step you will now come to the **Define Dimension(s)** page. Gauge charts are normally dimensionless, so this page is then automatically skipped.

Select one or more dimensions in the drop-down boxes. The dimension(s) define the values for which to calculate the chart expression(s). Dimensions are typically found to the left in table charts and on the x-axis in e.g. bar charts.

After finishing this wizard you can add more dimensions or change dimensions at any time via the **Chart Properties** dialog.

Click **Next** to continue.

Step 3 - Define Expression

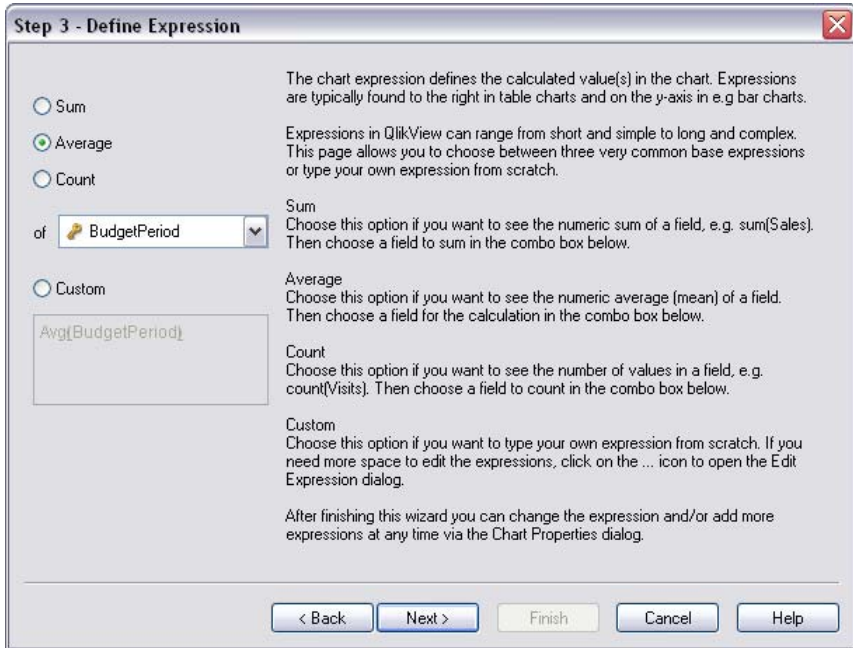


Figure 73. The third page of the quick chart wizard

The chart expression defines the calculated value(s) in the chart. Expressions are typically found to the right in tables and on the y-axis in e.g. bar charts.

Expressions in QlikView can range from short and simple to long and complex. This page allows you to choose between three very common base expressions or type your own expression from scratch.

Sum

Choose this option if you want to see the numeric sum of a field, e.g. `sum(Sales)`. Then choose a field to sum in the combo box below.

Average

Choose this option if you want to see the numeric average (mean) of a field, e.g. `avg(Score)`. then choose a field for the calculation in the combo box below.

Count

Choose this option if you want to see the number of values in a field, e.g. **count(OrderID)**. Then choose a field to count in the combo box below.

Custom

Choose this option if you want to type your own expression from scratch. If you need more space to edit the expressions, click on the ... icon to open the **Edit Expression** dialog.

After finishing this wizard you can change the expression and/or add more expressions at any time via the **Chart Properties** dialog.

Click **Next** to continue.

The fourth step in the wizard allows you to adjust the format of the chart. As format settings vary between chart types, the contents of the page will be different depending on which chart type you selected in Step 1 of the wizard.

Step 4 - Chart Format (Bar Chart)

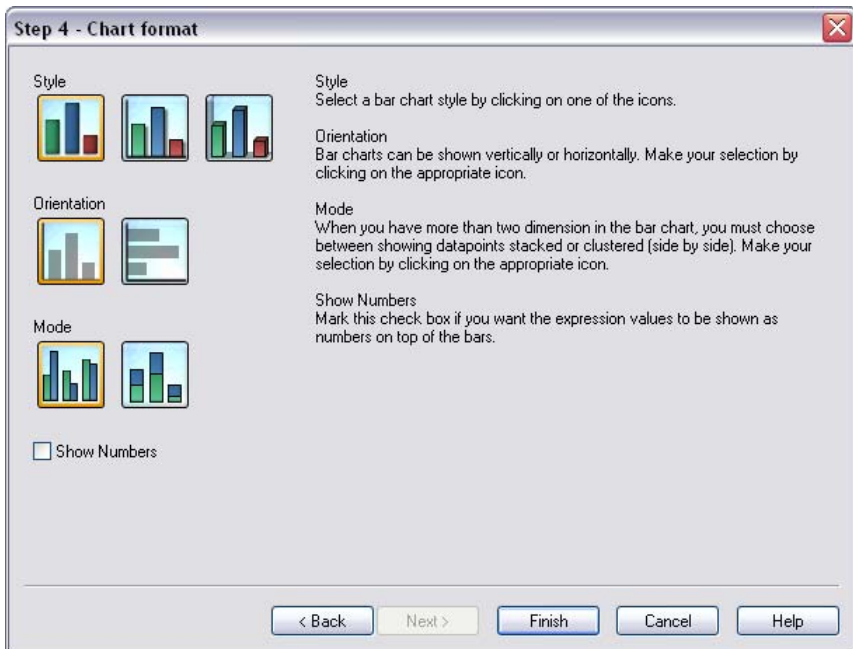


Figure 74. The fourth page of the quick chart wizard for bar charts

In this page you set formatting options for the selected chart type.

Style

Select a bar chart style by clicking on one of the icons.

Orientation

Bar charts can be shown vertically or horizontally. Make your selection by clicking on the appropriate icon.

Mode

When you have more than two dimensions in the bar chart you must choose between showing data points stacked or clustered (side by side). Make your selection by clicking the appropriate icon.

Show Numbers

Mark this check box if you want the expression values to be shown as numbers on top of the bars.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

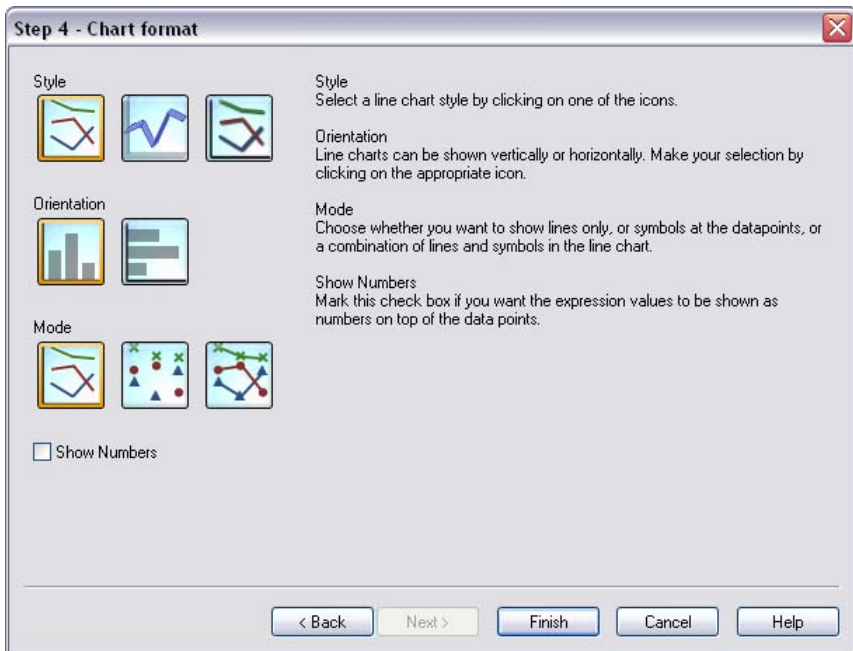
Step 4 - Chart Format (Line Chart)

Figure 75. The fourth page of the quick chart wizard for line charts

In this page you set formatting options for the selected chart type.

Style

Select a line chart style by clicking one of the icons.

Orientation

Line charts can be shown vertically or horizontally. Make your selection by clicking on the appropriate icon.

Mode

You may show data points as lines, symbols or a combination of thereof. Make your selection by clicking the appropriate icon.

Show Numbers

Mark this check box if you want the expression values to be shown as numbers on top of the data points.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

Step 4 - Chart Format (Pie Chart)

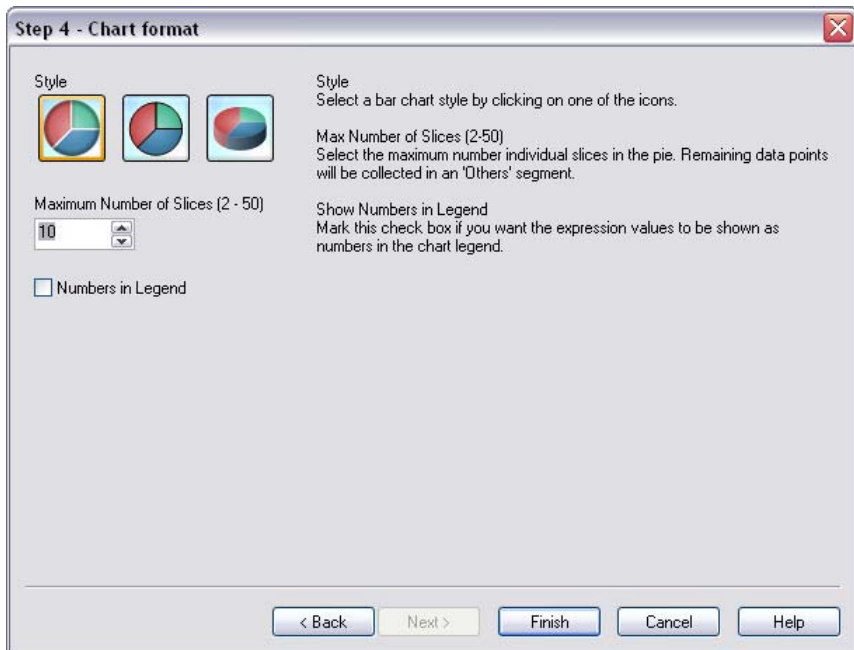


Figure 76. The fourth page of the quick chart wizard for pie charts

In this page you set formatting options for the selected chart type.

Style

Select a pie chart style by clicking one of the icons.

Max Number of Slices (2-50)

Select the maximum number of individual slices in the pie chart.
Remaining data points will be collected in the **Others** segment.

Show Numbers in Legend

Mark this check box if you want the expression values to be shown as numbers in the chart legend.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

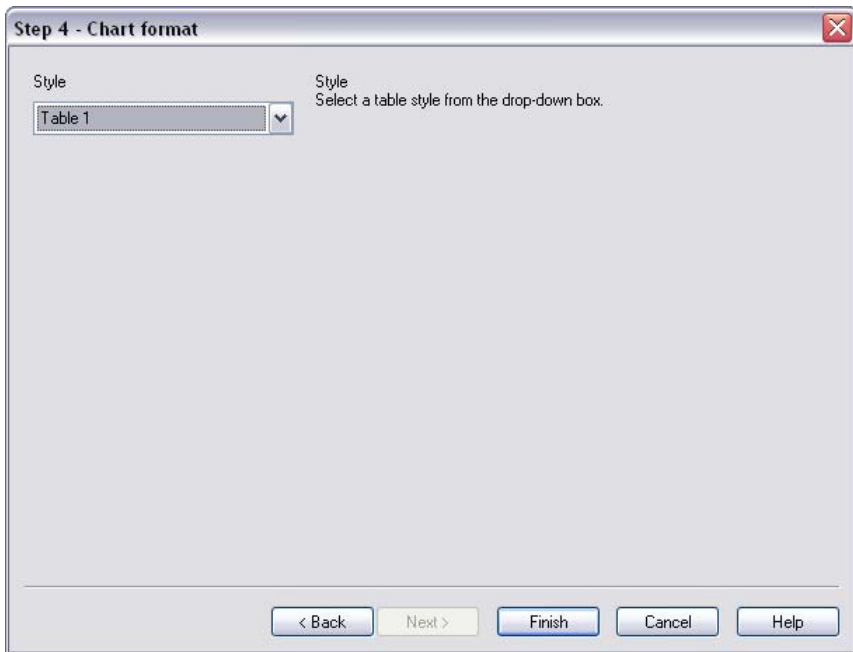
Step 4 - Chart Format (Straight Table)

Figure 77. The fourth page of the quick chart wizard for straight tables

In this page you set formatting options for the selected chart type.

Style

Select a table style from the drop-down box.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

Step 4 - Chart Format (Pivot Table)

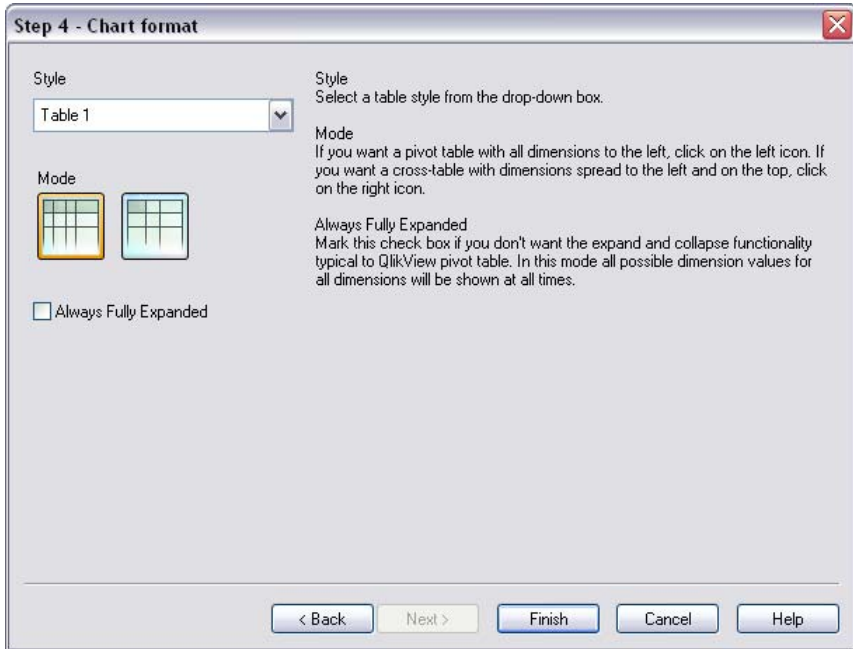


Figure 78. The fourth page of the quick chart wizard for pivot tables

In this page you set formatting options for the selected chart type.

Style

Select a table style from the drop-down box.

Mode

If you want a pivot table with all dimensions to the left, click the left icon. If you want a cross table with dimensions spread to the left and on the top, click the right icon.

Always Fully Expanded

Mark this check box if you do not want the expand and collapse functionality in the QlikView pivot table. In this mode all possible dimension values for all dimensions will be shown at all times.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

Step 4 - Chart Format (Gauge Chart)

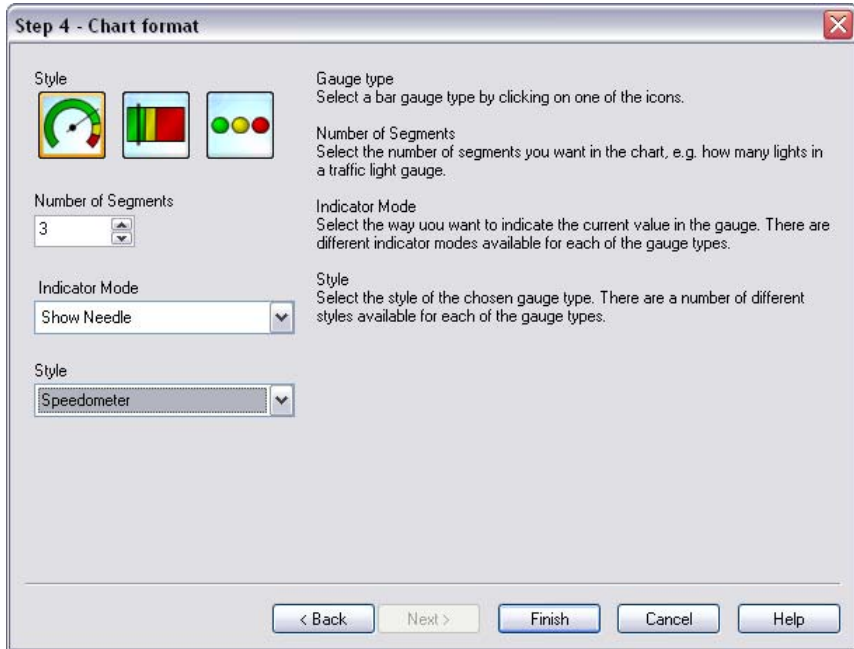


Figure 79. The fourth page of the quick chart wizard for gauge charts

In this page you set formatting options for the selected chart type.

Gauge Type

Select a bar gauge type by clicking one of the icons.

Number of Segments

Select the number of segments you want in the chart, e.g. how many lights in a traffic light gauge.

Indicator Mode

Select the way you want to indicate the current value in the gauge. There are different indicator modes available for each of the gauge types.

Style

Select the style of the chosen gauge type. There are a number of different styles available for each of the gauge types.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

17 THE TIME CHART WIZARD

The time chart wizard helps you to build charts where a given measure (expression) should be qualified and often compared by different time periods.


As an example you may want to show an expression such as `Sum(Sales)` but just for the last year or the current quarter to date. Often you will also want to compare that result with the same measure for a previous period, e.g. the year before or the same quarter last year. QlikView contains a number of functions to build expressions to do just this, but to the beginner they may seem hard to master. The time chart wizard has been designed to help you with the task of enclosing your basic measure expression in suitable time qualification functions according to your specifications.

The time chart wizard takes you through the following basic steps:

- 1 Select chart type
- 2 Define chart dimension(s)
- 3 Define the base expression and time references
- 4 Select time period
- 5 Define period offset
- 6 Define comparison period
- 7 Column selection and presentation

Depending on selections some of the above steps may be skipped.

17.1 Starting the Time Chart Wizard

You start the time chart wizard either by choosing **Time Chart Wizard** from the **Layout** menu or by clicking on the  button in the **Design** toolbar.

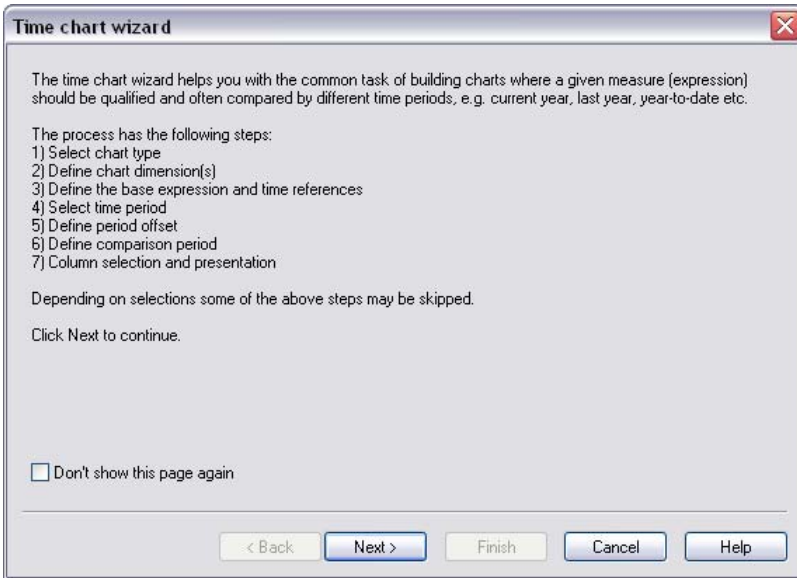


Figure 80. The start page of the Time chart wizard

The first time you start the wizard you will be met by a start page outlining the purpose of the wizard and the basic steps involved. If you want to skip the start page when you use the wizard in the future, mark the **Don't show this page again** check box.

Click **Next** to continue.

17.2 Pages of the Time Chart Wizard

Step 1 - Select Chart Type

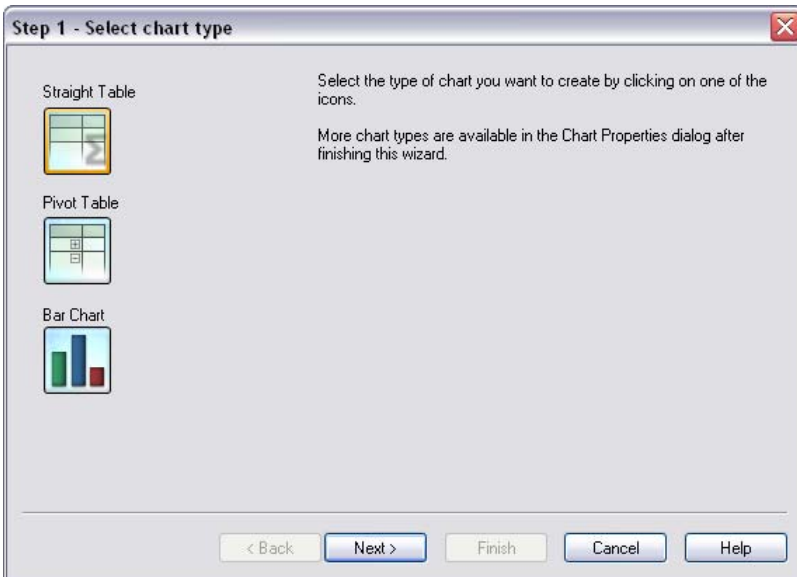


Figure 81. The first page of the Time chart wizard

Select the type of chart you want to create by clicking on one of the icons.

The chart types available are those typically used in charts involving time period qualification. You may change the chart into any other QlikView chart type via the **Chart Properties** dialog after finishing the wizard.

Click **Next** to continue.

Step 2 - Define dimension(s)

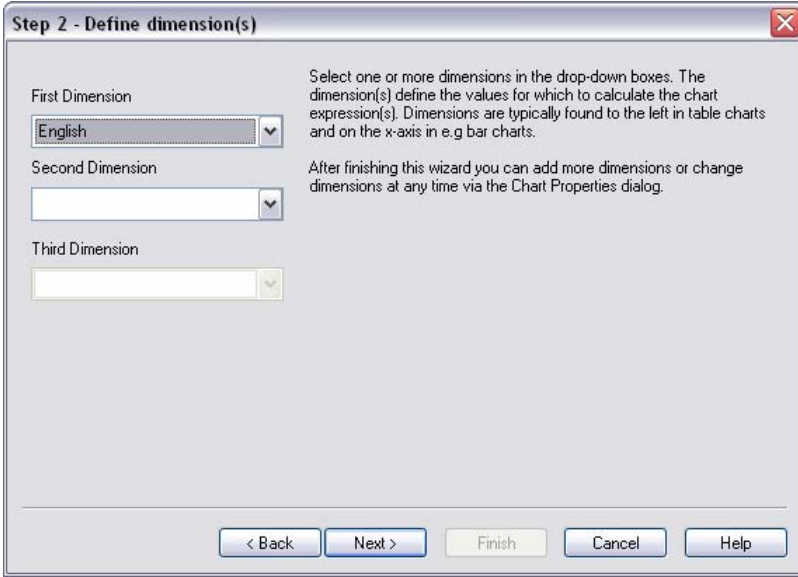


Figure 82. The second page of the time chart wizard

This page defines the chart's dimension(s).

Select one or more dimensions in the drop-down boxes. The dimension(s) define the values for which to calculate the chart expression(s). Dimensions are typically found to the left in table charts and on the x-axis in e.g. bar charts.

After finishing this wizard you can add more dimensions or change dimensions at any time via the **Chart Properties** dialog.

Click **Next** to continue.

Step 3 - Expression and Time References

Step 3 - Expression and time references

Base Expression
 ...
 Base Expression
 Type an expression defining the measure you want to study, e.g. sum(Sales)

Timestamp Expression
 ...
 Timestamp Expression
 State an expression (or just a field name) indicating the location of a timestamp (typically a date) associated with each transaction, e.g. OrderDate.

Basedate Expression
 ...
 Basedate Expression
 Type an expression providing the reference in time towards all transaction timestamps should be compared, e.g. today().

If you need more space to edit any of the expressions, click on the ... icon to open the Edit Expression dialog.

< Back Next > Finish Cancel Help

Figure 83. The third page of the Time chart wizard

This page defines the chart's underlying base expression and the time references.

Base Expression

Type an expression defining the measure you want to study, e.g. **sum(Sales)**.

Timestamp Expression

State an expression (or just a field name) indicating the location of a timestamp (typically a date) associated with each transaction e.g. *OrderDate*.

Basedate Expression

Type an expression providing the reference in time towards all transaction timestamps should be compared, e.g. **today()**.

If you need more space to edit any of the expressions, click on the ... icon to open the **Edit Expression** dialog.

Click **Next** to continue.

Step 4- Select Time Period

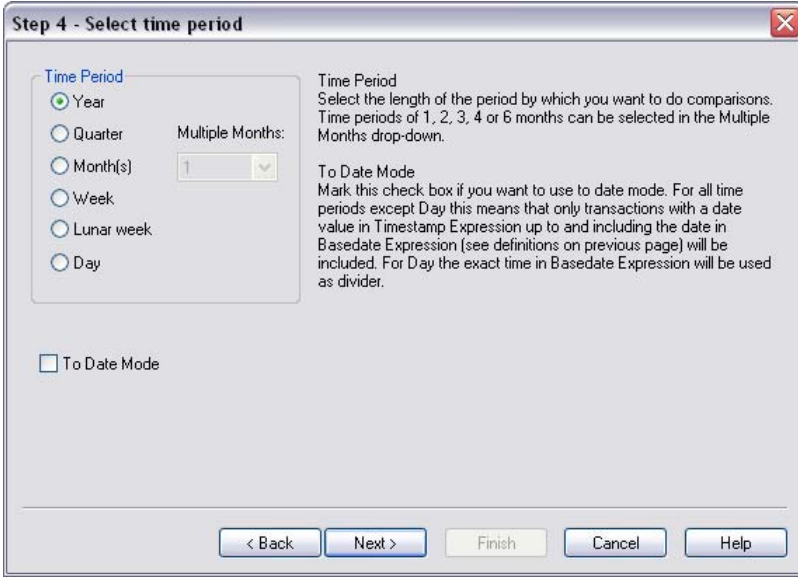


Figure 84. The fourth page of the Time chart wizard

On this page you select the basic time period unit for the chart.

Time Period

Select the length of the period by which you want to qualify the chart expression and do comparisons. When using the Month(s) option time periods of 1, 2, 3, 4, or 6 months can be selected in the Multiple Months drop-down. The value 3 can be given, but will correspond exactly to Quarter. Lunar weeks are defined as consecutive 7-day periods starting January 1st each year. There are always exactly 52 lunar weeks in a year. Lunar week 52 will thus contain 8 or 9 days.

To Date Mode

Mark this check box if you want to use date mode. For all time periods except **Day** this means that only transactions with a date value in **Timestamp Expression** up to and including the date in **Basedate Expression** (see definitions on previous page) will be included. For **Day** the exact time in **Basedate Expression** will be used as divider.

Click **Next** to continue.

Step 5 - Period Start Offset



Figure 85. The fifth page of the Time chart wizard

In this page you may state an offset to the beginning of the chosen time period. In many cases this page can be skipped. Depending on the **Time Period** chosen in step 4 above, one of the following two options will be given:

Year Start

If you work with (fiscal) years not starting in January, you may select the first month of your (fiscal) year in this drop-down. This option is only available if the basic **Time Period** chosen in step 4 above is **Year, Quarter of Months**.

Week/Day Offset

If you want to offset the start of the week or the start of the day you can do it here. Offsets are given in days and/or fractions thereof. E.g. 1 for calendar week starting Tuesday or 0.125 for day starting 3am. This option is only available if the basic **Time Period** chosen in step 4 above is **Week, Lunar Week or Day**.

Click **Next** to continue.

Step 6 - Define Comparison Period

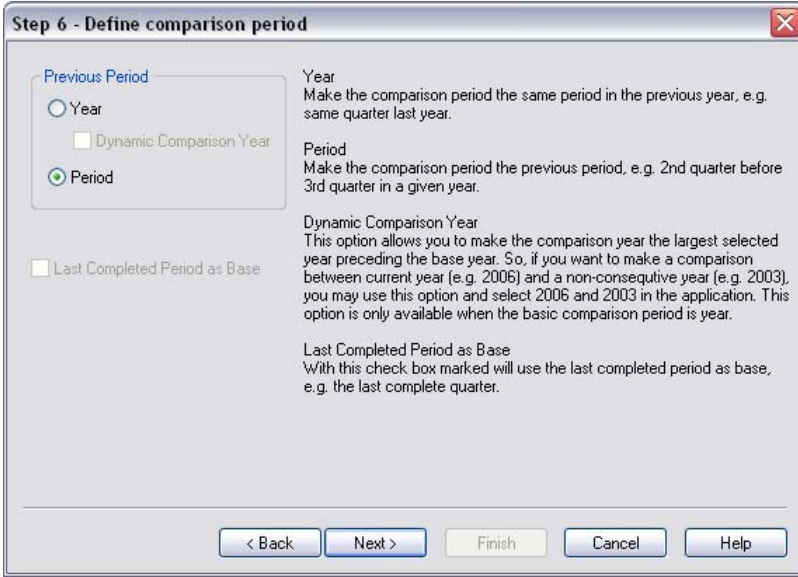


Figure 86. The sixth page of the Time chart wizard

In this page you define the comparison period. If you have chosen **Year** as main **Time Period** in step 4 above, this page is automatically skipped.

Year

Make the comparison period in the previous year, e.g. same quarter last year.

Period

Make the comparison period the previous period, e.g. 2nd quarter before 3rd quarter in a given year.

Dynamic Comparison Year

This option allows you to make the comparison year the largest selected year preceding the base year. So, if you want to make comparison between the current year (e.g. 2006) and a non-consecutive year (e.g. 2003), you may use this option and select 2006 and 2003 in the application. This option is only available when the basic comparison period is year.

Last Completed Period as Base

With this check box marked, the last completed period will be used as base, e.g. the last complete quarter.

Click **Next** to continue.

Step 7 - Column Selection and Presentation

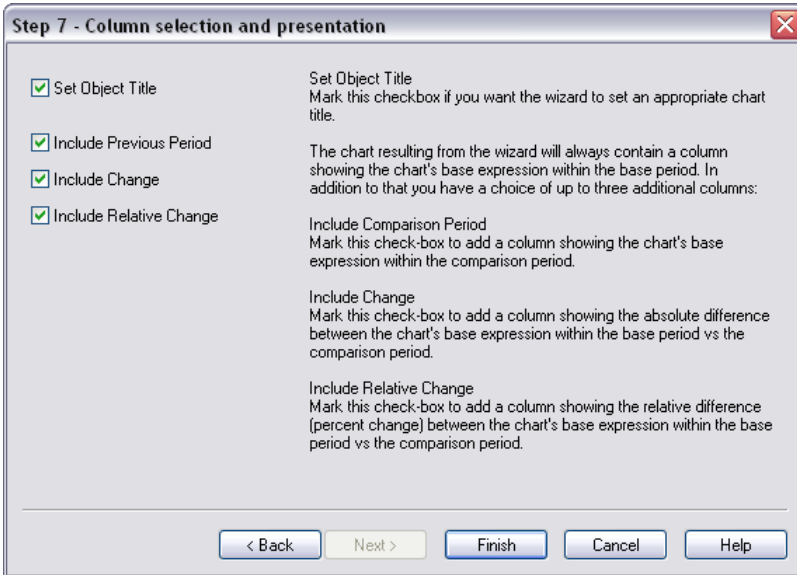


Figure 87. The final (7th) page of the Time chart wizard

In this page you choose which expression columns you want to include in the chart and whether you want the wizard to set the chart title.

Set Object Title

Mark this check box if you want the wizard to set an appropriate chart title.

The chart resulting from the wizard will always contain a column showing the chart's base expression within the base period. In addition to that you have a choice of up to three additional columns:

Include Comparison Period

Mark this check box to add a column showing the chart's base expression within the comparison period.

Include Change

Mark this check box to add a column showing the absolute difference within the comparison period.

Include Relative Change

Mark this check box to add a column showing the relative difference (percent change) between the chart's base expression within the base period vs. the comparison period.

Click **Finish** to finalize the chart and return to the layout. As mentioned before you may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

18 THE STATISTICS CHART WIZARD

The Statistics chart wizard provides guidance for those who want to apply common statistical tests on data in QlikView.

18.1 Starting the Statistics Chart Wizard

You start the statistics chart wizard by choosing **Statistics Chart Wizard** from the **Layout** menu.

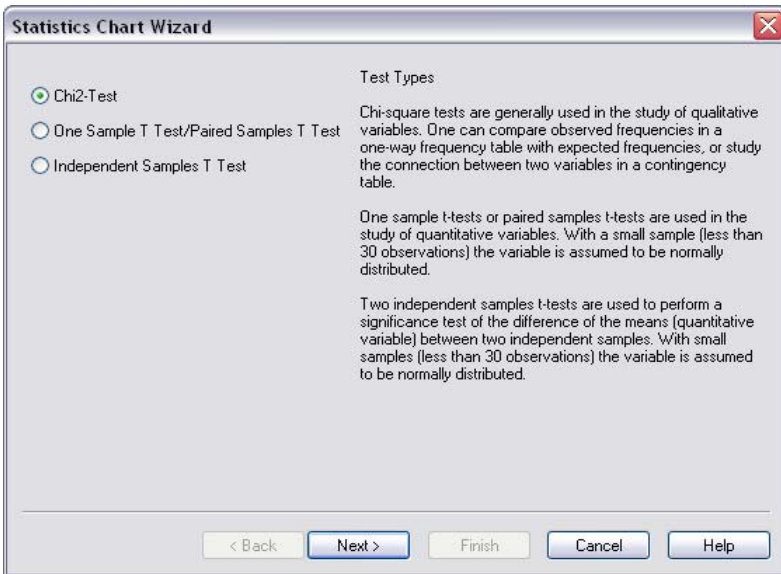


Figure 88. The start page of the Statistics chart wizard

The start page offers you a choice of statistical tests. Select the appropriate type of test and then click **Next** to continue.

Please note that QlikView supports more types of tests than those covered in the wizard. Please see the section on Statistical test functions on page 372.

For information about statistical tests and their uses, please refer to a statistics text book.

18.2 Chi2-Test

This option generates a straight table with values returned from a chi2 test. A chi2 test is often used to compare two sets of values and determine the probability that the two sets come from the same statistical distribution.

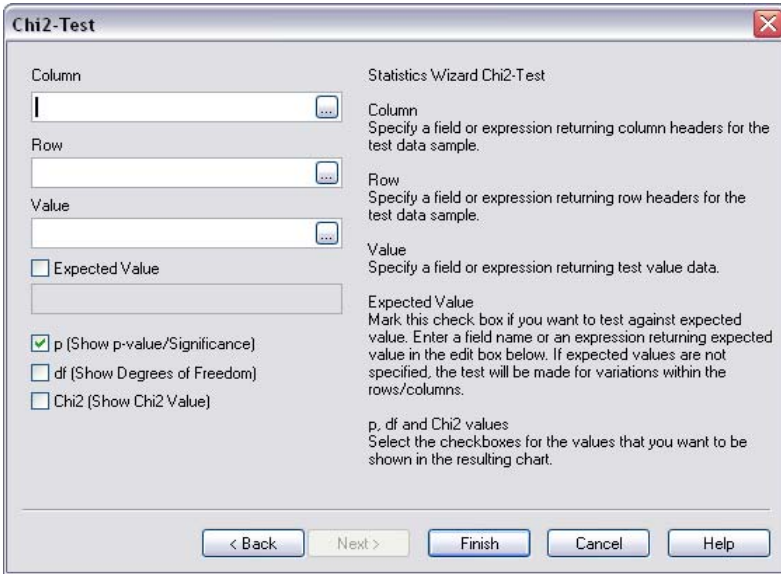


Figure 89. The chi2 test page of the statistics chart wizard

To obtain the chi2 test chart, you must specify the following:

Column

Specify a field or expression returning column headers for the test data sample.

Row

Specify a field or expression returning row headers for the test data sample.

Value

Specify a field or expression returning test value data.

Expected Value

Mark this check box if you want to test against expected value. Enter a field name or an expression returning expected value in the edit box below. If expected values are not specified the test will be made for variations within the rows/columns.

p (Show p-value / Significance)

Mark this check box to include the p value (significance) in the resulting chart.

df (Show Degrees of Freedom)

Mark this check box to include the df value (degrees of freedom) in the resulting chart.

Chi2 (Show Chi2 Value)

Mark this check box to include the test value in the resulting chart.

Click **Finish** to finalize the chart and return to the layout. You may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

18.3 Paired t-test

This option generates a straight table with values returned from a paired student's t-test.

Figure 90. The paired t-test page of the Statistics chart wizard

To obtain the t-test chart, you must specify the following:

Value Field/Expression

Specify a field or expression returning test value data series.

Test Value

Specify a field or expression returning a test value.

t (Show t-value)

Mark this check box to include the t value in the resulting chart.

df (Show Degrees of Freedom)

Mark this check box to include the df value (degrees of freedom) in the resulting chart.

p (Show p-value / Significance (2-tailed))

Mark this check box to include the p value (significance, 2-tailed) in the resulting chart.

Mean Difference

Mark this check box to include the mean difference value in the resulting chart.

Lower (CL%)

Mark this check box to include the lower limit of the confidence interval (as specified in **Confidence Level** below) in the resulting chart.

Upper (CL%)

Mark this check box to include the upper limit of the confidence interval (as specified in **Confidence Level** below) in the resulting chart.

Confidence Level

Specify a percentage for the level of confidence.

Click **Finish** to finalize the chart and return to the layout. You may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

18.4 Independent Samples t-test

This option generates a straight table with values returned from a two independent samples' student's t-test.

Figure 91. The independent samples t-test page of the Statistics chart wizard

To obtain the t-test chart, you must specify the following:

Grouping Field/Expression

Specify a field or expression returning test value data series. Exactly two values must be returned.

Test Field/Expression

Specify a field or expression returning the test value data series.

t (Show t-value)

Mark this check box to include the t value in the resulting chart.

df (Show Degrees of Freedom)

Mark this check box to include the df value (degrees of freedom) in the resulting chart.

p (Show p-value / Significance (2-tailed))

Mark this check box to include the p value (significance, 2-tailed) in the resulting chart.

Mean Difference

Mark this check box to include the mean difference value in the resulting chart.

Std. Error Difference

Mark this check box to include the standard error difference value in the resulting chart.

Lower (CL%)

Mark this check box to include the lower limit of the confidence interval (as specified in **Confidence Level** below) in the resulting chart.

Upper (CL%)

Mark this check box to include the upper limit of the confidence interval (as specified in **Confidence Level** below) in the resulting chart.

Confidence Level

Specify a percentage for the level of confidence.

Click **Finish** to finalize the chart and return to the layout. You may return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

19 CHART EXPRESSIONS

Expressions are used in charts to define the calculated data, e.g. the data shown on the y-axis in a standard bar or line chart. In most cases it is enough to choose a pre-defined aggregation function in the Edit Expression dialog and automatically generate an expression. QlikView supports very complex expressions, involving one or more fields and most of the functions available in the script expression syntax. Please note that some differences exist between script expressions and chart expressions in terms of syntax and available functions.

Note The most important difference between script and chart expressions is the role of the aggregation functions and the use of field references. The basic rule is that any field name used in a chart expression must be enclosed by exactly one aggregation function. This means that: 1) A field name or an expression containing a field name must always be enclosed by an aggregation function. 2) An aggregation function must never have as argument an expression containing an aggregation function.

Note In a calculated formula, field names may optionally be used without an enclosing aggregation function. In that case, **only** will be used as aggregation function.

All expressions in QlikView return a number and/or a string, or a null value, whichever is appropriate. Logical functions and operators return 0 for false and -1 for true. Number to string conversions and vice versa are implicit. Logical operators and functions interpret 0 as false and all else as true. Expressions which cannot be correctly evaluated, e.g. as a result of incorrect parameters to functions, return null.

19.1 Chart Expression Syntax

The general syntax for a chart expression is:

```

expression ::= (
                    constant |
                    expressionname |
                    operator1 expression |
                    expression operator2 expression |
                    function |
                    aggregationfunction |
                    (expression ) )
  
```

where

constant is a string (a text, a date or a time) enclosed by single straight quotation marks, or a number. Constants are written with no thousands separator and with a decimal point as decimal separator.

expressionname is the name (label) of another expression in the same chart.

operator1 is a unary operator (working on one expression, the one to the right).

operator2 is a binary operator (working on two expressions, one on each side).

function ::= functionname (parameters)

parameters ::= expression { , expression }

The number and types of parameters is not arbitrary. It depends on the function used.

aggregationfunction ::= aggregationfunctionname (parameters2)

parameters2 ::= aggregexpr { , aggregexpr }

The number and types of parameters is not arbitrary. It depends on the function used.

*aggregexpr ::= (fieldref |
operator1 aggregexpr |
aggregexpr operator2 aggregexpr |
functioninaggr |
(aggregexpression))*

fieldref is a field name.

functioninaggr ::= functionname (parameters2)

Expressions and functions can thus be nested freely, and as long as a *fieldref* is always enclosed by exactly one aggregation function. As long as the expression returns an interpretable value, QlikView will not give any error messages.

19.2 Variable Text Expansion

As from QlikView version 6.04 QlikView variables can be used for text expansion inside an expression in a QlikView chart or elsewhere in the QlikView layout. When used for this purpose, the variable's text value is inserted into the expression before the expression is evaluated.

When using a variable for text expansion inside a statement, the syntax

$\$(\text{variablename})$

is used. $\$(\text{variablename})$ expands to the value in *variablename*. If *variablename* does not exist the expansion will be the empty string.

For numeric text expansion the syntax

$\$(\# \text{variablename})$

is used. $\$(\# \text{variablename})$ always yields a legal decimal-point number reflection of the numeric value of *variablename*, possibly with exponential notation (for very large/small) numbers). If *variablename* does not exist or does not contain a numeric value, it will be expanded to 0 instead.

The numeric expansion is probably considerably less applicable than text expansion when used in chart expressions, but is nevertheless documented here for the sake of completeness.

Example:

The variable *x* contains the text string *sum(Sales)*.

In a chart you define the expression $\$(x)/12$. The effect is exactly the same as if having the chart expression *sum(Sales)/12*.

However, if you change the value of the variable *x* to *sum(Budget)*, the chart will immediately be recalculated with the expression interpreted as *sum(Budget)/12*.

19.3 Operators

There are two types of operators in QlikView, unary operators that take only one operand, and binary operators that take two operands. Most operators are binary.

Numeric Operators

All numeric operators use the numeric values of the operands and return a numeric value as result.

+

Sign for positive number (unary operator) or arithmetic addition.
The binary operation returns the sum of the two operands.

-

Sign for negative number (unary operator) or arithmetic subtraction.
The unary operation returns the operand multiplied by -1, and the binary the difference between the two operands.

*

Arithmetic multiplication. The operation returns the product of the two operands.

/

Arithmetic division. The operation returns the ratio between the two operands.

String Operators

Only two string operators exist.

&

String concatenation. The operation returns a text string that consists of the two operand strings, one after another.

Example:

'abc' & 'xyz' returns 'abcxyz'

like

String comparison with wildcard characters. The operation returns a boolean true (-1) if the string before the operator is matched by the string after the operator. The second string may contain the wildcard characters * (any number of arbitrary characters) or ? (one arbitrary character).

Example:

'abc' **like** 'a*' returns true (-1)

'abcd' **like** 'a?c*' returns true (-1)

'abc' **like** 'a??bc' returns false (0)

Logical Operators

All logical operators interpret the operands logically and return true (-1) or false (0) as result.

not

Logical inverse. One of the few unary operators. The operation returns the logical inverse of the operand.

and

Logical and. The operation returns the logical and of the operands.

or

Logical or. The operation returns the logical or of the operands.

xor

Logical exclusive or. The operation returns the logical exclusive or of the operands. I.e. like logical or, but with the difference that the result is false if both operands are true.

Relational Operators

All relational operators compare the values of the operands and return true (-1) or false (0) as result. All relational operators are binary.

<

Less than. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

<=

Less than or equal. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

>

Greater than. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

>=

Greater than or equal. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

=

Equals. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

<>

Not equivalent to. A numeric comparison is made if both operands can be interpreted numerically. The operation returns the logical value of the evaluation of the comparison.

follows

ASCII greater than. Returns true if the left-hand operand has a text representation which, in an ASCII comparison, comes after the text representation of right-hand operand. Unlike the > operator no attempt is made to make a numeric interpretation of the argument values before the comparison.

Examples:

'23' follows '111' returns true

'23' > '111' returns false

precedes

ASCII less than. Returns true if the left-hand operand has a text representation which, in an ASCII comparison, comes before the text

representation of the right-hand operand. Unlike the < operator no attempt is made to make a numeric interpretation of the argument values before the comparison.

Examples:

'11' **precedes** '2' returns true

'11' < '2' returns false

Bit operators

All bit operators convert the operands to signed integers (32 bit) and return the result in that way. All operation are performed bit by bit.

bitnot

Bit inverse. Unary operator. The operation returns the logical inverse of the operand performed bit by bit.

bitand

Bit and. The operation returns the logical and of the operands performed bit by bit.

bitor

Bit or. The operation returns the logical or of the operands performed bit by bit.

bitxor

Bit exclusive or. The operation returns the logical exclusive or of the operands performed bit by bit.

>>

Bit right shift. Unary operator. The operation returns the operand shifted one step to the right.

<<

Bit left shift. Unary operator. The operation returns the operand shifted one step to the left.

19.4 Aggregation Functions

For a better overview the aggregation functions are grouped as follows:

Basic Aggregation Functions	page 299
String Aggregation Functions.....	page 303
Counter Aggregation Functions	page 305
Statistical Aggregation Functions.....	page 309
Financial Aggregation Functions	page 326
Statistical Test Functions	page 330

Special Input Field Aggregation Function page 353
 Special Function for Advanced Aggregations..... page 353
 Set Analysis page 353

Apart from the aggregation functions there are other functions that can be used in charts and layout. They are listed and explained on page 362. The chart aggregation functions can be used on fields in chart expressions. The argument expression of one aggregation function must not contain another aggregation function.

Basic Aggregation Functions

sum ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression*)

Returns the aggregated sum of *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded. If the word **distinct** occurs before a *field*, duplicates of field values will be disregarded.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **total** qualifier may be followed by a list of one or more field names within angle brackets. These field names should be a subset of the chart dimensions. In this case the calculation will be made disregarding all chart dimensions except those listed, i.e. one value will be returned for each combination of field values in the listed dimension fields. Also fields which are not currently a dimension in a chart may be included in the list. This may be useful in the case of group dimensions, where the dimension fields are not fixed. Listing all of the dimensions in the group causes the function to work when the cycle or drill-down level changes.

In previous QlikView versions, the **all** qualifier may occur before an expression. This is equivalent to using **{1} total**, i.e. in such a case, the calculation will be made over all the values of the field in the document, disregarding the chart dimensions and current selections. (The same value is always returned regardless of the logical state in the document.) If the **all** qualifier is used, a set expression cannot be used, since the **all** qualifier defines a set by itself. For legacy reasons, the **all** qualifier will still work in this QlikView version, but may be removed in coming versions.

Examples:

sum(Sales)

sum(Price * Quantity)

sum(distinct Price)

sum(Sales) / sum(total Sales)

returns sales within selection

sum(Sales) / sum(total <Month> Sales)

returns sales within selection for each Month

sum(Sales) / sum(total <Month,Grp> Sales)

returns sales within selection for each Month and Grp

sum(Sales) / sum(total <Qtr,Month,Week> Sales)

possible syntax for use with a time drill-down group

sum(Sales) / sum({1} Total Sales)

returns sales within total document

sum({ BM01 } Sales)

returns sales within the selection defined by bookmark
BM01

sum({ \$<Year={ 2007,2008 }>}Sales)

returns the sales for the current selection but just for year
2007 and 2008, i.e. the same as **Sum(if(Year=2007 or
Year=2008,Sales))**

Further examples of aggregate qualifiers are found on page 449.

min ([set_expression] [distinct] [total [<fld {, fld}>]]expression [, rank]))

Returns the numeric minimum value of *expression* or *field* iterated over the chart dimension(s). *Rank* defaults to 1 which corresponds to the lowest value. By specifying *rank* as 2 the second lowest value will be returned. If *rank* is 3 the third lowest value will be returned and so on.

Expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced

nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

The **min** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

```
min( Sales )
min( Sales, 2)
min( Price*Quantity )
min( total Sales )
min( {1} total Sales )
min( total <Country> Sales )
```

Further examples of aggregate qualifiers are found on page 449.

max ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression* [, *rank*])

Returns the numeric maximum value of *expression* or field iterated over the chart dimension(s). *Rank* defaults to 1 which correspond to the highest value. By specifying *rank* as 2 the second highest value will be returned. If *rank* is 3 the third highest value will be returned and so on.

Expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

The **max** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

```
max( Sales )
max( Sales, 2)
max( Price*Quantity )
max( total Sales )
max( {1} total Sales )
max( total <Country> Sales )
```

Further examples of aggregate qualifiers are found on page 449.

only ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression*)

If *expression* or *field* iterated over the chart dimension(s) contains one single value, that value is returned, else NULL is returned. **Only** can return numeric values as well as text values. *Expression* must not contain aggregation functions.

The word **distinct** before an *expression* or a *field* is allowed but has no meaning.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **only** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

only(Sales)

only(Price*Quantity)

only(**total** Salesman)

only({1} **total** Sales)

only(**total** <District> Salesman)

Further examples of aggregate qualifiers are found on page 449.

mode([*set_expression*] [**distinct**] *expression*)

Returns the mode value, i.e. the most commonly occurring value, of *expression* or *field* iterated over the chart dimension(s). If more than one value is equally commonly occurring, NULL is returned. **Mode** can return numeric values as well as text values. *Expression* must not contain aggregation functions.

The word **distinct** before an *expression* or a *field* is allowed but has no meaning.

Examples:

mode(Product)

mode(X*Y/3)

firstsortedvalue ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>] *expression*)

returns the first value of *expression* sorted by corresponding *sort-weight* when *expression* is iterated over the chart dimension(s). *Sort-weight* should return a numeric value where the lowest value will render the corresponding value of *expression* to be sorted first. By preceding the *sort-value* expression with a minus sign, the function will return the last value instead. If more than one value of *expression* share the same lowest *sort-order*, the function will return Null. By stating an *n* larger than 1, you will get the *n*th value in order.

The function argument expressions must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

The word **distinct** before *expression* or a field is allowed but has no meaning.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **firstsortedvalue** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

firstsortedvalue (PurchasedArticle, OrderDate)

firstsortedvalue (PurchasedArticle, OrderDate, -2)

firstsortedvalue (A/B, X*Y/3)

firstsortedvalue (**distinct** PurchasedArticle, OrderDate)

firstsortedvalue (**total** PurchasedArticle, OrderDate)

firstsortedvalue (**total** <Grp> PurchasedArticle, OrderDate)

String Aggregation Functions

MinString([*set_expression*] [**total** [<*fld* {, *fld*>]] *expression*)

If *expression* or *field* iterated over the chart dimension(s) contains one or more values with a string representation (any text or number), the first value in text sort order is returned, else NULL is returned.

The function argument expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **total** occurs before expression the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **minstring** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

MinString(Currency)

MinString(Left(abc,2))
MinString(total Currency)
MinString(total <X> Currency)

MaxString([set_expression][total [<fld {, fld}>]] expression)

If *expression* or *field* iterated over the chart dimension(s) contains one or more values with a string representation (any text or number), the last value in text sort order is returned, else NULL is returned.

The function argument *expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **total** occurs before *expression* the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **maxstring** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

MaxString(Currency)
MaxString(Left(abc,2))
MaxString(total Currency)
MaxString(total <X> Currency)

concat ([set_expression][distinct] [total [<fld {, fld}>]] expression)

returns the aggregated string concatenation of all values of *expression* iterated over the chart dimension(s). Each value may be separated by the string found in *delimiter*. The order of concatenation may be determined by *sort-weight*. *Sort-weight* should return a numeric value where the lowest value will render the item to be sorted first.

The function argument *expressions* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **distinct** occurs before *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **concat** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

```
concat( Code, ';' )
concat( FirstName&' '&LastName, ';' )
concat( distinct Code, ';' )
concat( total Name, ';' , Date )
concat( total <Grp> Name, ';' , Date)
```

Counter Aggregation Functions

```
count( [set_expression][ distinct ] [ total [<fld {, fld}>] ]
expression )
```

Returns the aggregated total count of values from *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before a *field*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **count** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

```
count( Sales )
count( Price*Quantity )
count( distinct Price )
count( Sales ) / count( total Sales )
    returns share within selection
count( Sales ) / count( total <Month> Sales )
    returns share within selection for each Month
count( Sales ) / count( total <Month,Grp> Sales )
    returns share within selection for each Month and Grp
```

count(Sales) / count(total <Qtr,Month,Week> Sales)
possible syntax for use with a time drill-down group

count(Sales) / count({1} total Sales)
returns share within total document

count(Sales) / count(distinct {1} total Sales)

Further examples of aggregate qualifiers are found on page 449.

NumericCount ([set_expression] [distinct] [total [<fld {, fld}>]] expression | field)

Returns the aggregated numeric count of values from *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **numericcount** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

NumericCount(Sales)

NumericCount(Price*Quantity)

NumericCount(distinct Price)

NumericCount(Sales) / NumericCount(total Sales)

returns share within selection

NumericCount(Sales) / NumericCount(total <Month> Sales)

returns share within selection for each Month

NumericCount(Sales) / NumericCount(total <Month,Grp> Sales)

returns share within selection for each Month and Grp

NumericCount(Sales) / NumericCount(total <Qtr,Month,Week> Sales)

possible syntax for use with a time drill-down group

NumericCount(Sales) / NumericCount({1} total Sales)

returns share within total document

Further examples of aggregate qualifiers are found on page 449.

TextCount ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression* | *field*)

Returns the aggregated text count of values from *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **textcount** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

TextCount(Sales)

TextCount(Price*Quantity)

TextCount(**distinct** Price)

TextCount(Sales) / **TextCount**(**total** Sales)

returns share within selection

TextCount(Sales) / **TextCount**(**total** <Month> Sales)

returns share within selection for each Month

TextCount(Sales) / **TextCount**(**total** <Month,Grp> Sales)

returns share within selection for each Month and Grp

TextCount(Sales) / **TextCount**(**total** <Qtr,Month,Week> Sales)

possible syntax for use with a time drill-down group

TextCount(Sales) / **TextCount**({1} **total** Sales)

returns share within total document

Further examples of aggregate qualifiers are found on page 449.

NullCount ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression* | *field*)

Returns the aggregated count of null values from *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **nullcount** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:

NullCount(Sales)

NullCount(Price*Quantity)

NullCount(distinct Price)

NullCount(Sales) / NullCount(total Sales)

returns share within selection

NullCount(Sales) / NullCount(total <Month> Sales)

returns share within selection for each Month

NullCount(Sales) / NullCount(total <Month,Grp> Sales)

returns share within selection for each Month and Grp

NullCount(Sales) / NullCount (total <Qtr,Month,Week> Sales)

possible syntax for use with a time drill-down group

NullCount(Sales) / NullCount({1} total Sales)

returns share within total document

Further examples of aggregate qualifiers are found on page 449.

MissingCount ([set_expression] [distinct] [total [<fld {, fld}>]] expression | field)

Returns the aggregated count of missing values from *expression* or *field* iterated over the chart dimension(s). Missing values are all non-numeric values, including null values. *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **missingcount** function supports sets and the **total** qualifier as described under the **sum** aggregation function, page 299.

Examples:**MissingCount**(Sales**MissingCount**(if(Price>10, Price, 'invalid'))**MissingCount**(distinct Price)**MissingCount**(Sales) / **MissingCount**(total Sales)

returns share within selection

MissingCount(Sales) / **MissingCount**(total <Month> Sales)

returns share within selection for each Month

MissingCount(Sales) / **MissingCount**(total <Month,Grp> Sales)

returns share within selection for each Month and Grp

MissingCount(Sales) / **MissingCount**(total <Qtr,Month,Week> Sales)

possible syntax for use with a time drill-down group

MissingCount(Sales) / **MissingCount**({1} total Sales)

returns share within total document

Further examples of aggregate qualifiers are found on page 449.

Statistical Aggregation Functions

avg ([set_expression] [distinct] [total [<fld {, fld}>]] expression |field)

Returns the aggregated average of *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of record can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before an *expression*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **total** qualifier may be followed by a list of one or more field names within angle brackets. These field names should be a subset of the chart dimensions. In this case the calculation will be made disregarding all chart dimensions except those listed, i.e. one value will be returned for each combination of field values in the listed

dimension fields. Also fields which are not currently a dimension in a chart may be included in the list. This may be useful in the case of group dimensions, where the dimension fields are not fixed. Listing all of the dimensions in the group causes the function to work when the cycle or drill-down level changes.

In previous QlikView versions, the **all** qualifier may occur before an expression. This is equivalent to using “**{1} total**”, i.e. in such a case, the calculation will be made over all the values of the field in the document, disregarding the chart dimensions and current selections. (The same value is always returned regardless of the logical state in the document.) For legacy reasons, **all** will still work. However, if the **all** qualifier is used, a set expression cannot be used, since the **all** qualifier defines a set by itself.

Examples:

avg(Sales)

avg(X*Y/3)

avg(**distinct** Price)

avg(**total** Sales)

avg(**{1} total** Sales)

avg(**total** <Grp> Price)

Further examples of aggregate qualifiers are found on page 449.

stdev([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *expression* \ *field*)

Returns the aggregated standard deviation of *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before a *field*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **stdev** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

```
stdev( Sales )
stdev( X*Y/3 )
stdev( distinct Price )
stdev( total Sales )
stdev( {1} total Sales )
stdev( total <Grp> Sales )
```

Further examples of aggregate qualifiers are found on page 449.

median ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression*)
returns the aggregated median of *expression* iterated over the chart dimension(s).

Expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **median** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

```
median( X )
median( X*Y/3 )
median( total X )
median( total <Group> Price )
```

fractile ([*set_expression*] [**distinct**] [**total** [<*fld* {, *fld*>]] *expression*,
fractile)

returns the aggregated *fractile* of *expression* iterated over the chart dimension(s).

The function argument expressions must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** func-

tion (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **fractile** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

fractile(Sales, 0.75)

fractile(X*Y/3, 0.9)

fractile(**total** Price, 0.25)

fractile(**total** <Group> Price)

skew([*set_expression*][**distinct**] [**total** [<*fld* {, *fld*>]] *expression* | *field*)

Returns the aggregated skewness of *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 459) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before a *field*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **skew** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:**skew**(Sales)**skew**(X*Y/3)**skew**(**distinct** Price)**skew**(**total** Sales)**skew**({1} **total** Sales)**skew**(**total** <Grp> Sales)

Further examples of aggregate qualifiers are found on page 449.

kurtosis([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *expression* | *field*)

Returns the aggregated kurtosis of *expression* or *field* iterated over the chart dimension(s). *Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 459) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before a *field*, the calculation will be made over all possible values given the current selections, but disregarding the chart dimensions.

The **kurtosis** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:**kurtosis**(Sales)**kurtosis**(X*Y/3)**kurtosis**(**distinct** Price)**kurtosis**(**total** Sales)**kurtosis**({1} **total** Sales)**kurtosis**(**total** <Grp> Sales)

Further examples of aggregate qualifiers are found on page 449.

correl ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *x-expression*, *y-expression*)

returns the aggregated correlation coefficient for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded. See also the **rangecorrel** function (page 373).

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates result from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **correl** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

correl(Payments, Dates)

correl(A/B, X*Y/3)

correl(**total** Payments, Dates)

correl(**total** <Grp> Payments, Dates)

sterr ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *expression*)

returns the aggregated standard error (stdev/sqrt(n)) for a series of values represented by *expression* iterated over the chart dimension(s). Text values, null values and missing values will be disregarded.

Expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **distinct** occurs before the function argument, duplicates resulting from the evaluation of the function argument will be disregarded.

If the word **total** occurs before the function argument the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **sterr** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

- sterr(X)**
- sterr(X*Y/3)**
- sterr(total X)**
- sterr(total <Z> X)**

steyx ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression*)

returns the aggregated standard error of the predicted y-value for each x-value in the regression for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **steyx** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

steyx(Y, X)

steyx(A/B, X*Y/3)

steyx(**total** Y, X)

steyx(**total** <Z> Y, X)

linest_m ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated m value (slope) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_m** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

- linest_m**(Y, X)
- linest_m**(A/B, X*Y/3)
- linest_m**(**total** Y, X)
- linest_m**(**total** <Z> Y, X)
- linest_m**(Y, X, 0)
- linest_m**(Y, X, 1, 1)

linest_b ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated b value (y-intercept) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_b** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

linest_b(Y, X)

linest_b(A/B, X*Y/3)

linest_b(**total** Y, X)

linest_b(**total** <Z> Y, X)

linest_b(Y, X, 0)

linest_b(Y, X, 1, 1)

linest_r2 ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated r2 value (coefficient of determination) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_r2** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

- linest_r2**(Y, X)
- linest_r2**(A/B, X*Y/3)
- linest_r2**(**total** Y, X)
- linest_r2**(**total** <Z> Y, X)
- linest_r2**(Y, X, 0)
- linest_r2**(Y, X, 1, 1)

linest_sem ([*set_expression*] [**distinct**] [**total** [<fld [, fld]>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated standard error of the m value of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_sem** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

linest_sem(Y, X)
linest_sem(A/B, X*Y/3)
linest_sem(**total** Y, X)
linest_sem(**total** <Z> Y, X)
linest_sem(Y, X, 0)
linest_sem(Y, X, 1, 1)

linest_seb ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated standard error of the b value of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_seb** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

```
linest_seb( Y, X )
linest_seb( A/B, X*Y/3 )
linest_seb( total Y, X )
linest_seb( total <Z> Y, X )
linest_seb( Y, X, 0 )
linest_seb( Y, X, 1, 1 )
```

linest_sey ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated standard error of the y estimate of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_sey** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

linest_sey(Y, X)

linest_sey(A/B, X*Y/3)

linest_sey(**total** Y, X)

linest_sey(**total** <Z> Y, X)

linest_sey(Y, X, 0)

linest_sey(Y, X, 1, 1)

linest_df ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated degrees of freedom of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_df** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

- linest_df**(Y, X)
- linest_df**(A/B, X*Y/3)
- linest_df**(**total** Y, X)
- linest_df**(**total** <Z> Y, X)
- linest_df**(Y, X, 0)
- linest_df**(Y, X, 1, 1)

linest_f ([*set_expression*] [**distinct**] [**total** [*<fld {, fld}>*]] *y-expression*, *x-expression* [, *y0* [, *x0*]]

returns the aggregated F statistic ($r^2/(1-r^2)$) of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_f** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

linest_f(Y, X)

linest_f(A/B, X*Y/3)

linest_f(**total** Y, X)

linest_f(**total** <Z> Y, X)

linest_f(Y, X, 0)

linest_f(Y, X, 1, 1)

linest_sreg ([*set-expression*] [**distinct**] [**total** [<fld [, fld]>]] *y-expression*, *x-expression* [, *y0* [, *x0*]])

returns the aggregated regression sum of squares of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_ssreg** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

- linest_ssreg**(Y, X)
- linest_ssreg**(A/B, X*Y/3)
- linest_ssreg**(**total** Y, X)
- linest_ssreg**(**total** <Z> Y, X)
- linest_ssreg**(Y, X, 0)
- linest_ssreg**(Y, X, 1, 1)

linest_ssresid ([*set_expression*] [**distinct**] [**total** [*<fld {, fld}>*]] *y-expression, x-expression* [, *y0* [, *x0*]])

returns the aggregated residual sum of squares of a linear regression defined by the equation $y=mx+b$ for a series of coordinates represented by paired numbers in *x-expression* and *y-expression* iterated over the chart dimension(s). Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair to be disregarded.

An optional value *y0* may be stated forcing the regression line to pass through the y-axis at a given point. By stating both *y0* and *x0* it is possible to force the regression line to pass through a single fixed coordinate.

Unless both *y0* and *x0* are stated, the function requires at least two valid data-pairs to calculate. If *y0* and *x0* are stated, a single data pair will do.

X-expression and *y-expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **linest_ssresid** function supports sets and the **total** qualifier as described under the **avg** aggregation function, page 309.

Examples:

linest_ssresid(Y, X)

linest_ssresid(A/B, X*Y/3)

linest_ssresid(**total** Y, X)

linest_ssresid(**total** <Z> Y, X)

linest_ssresid(Y, X, 0)

linest_ssresid(Y, X, 1, 1)

Financial Aggregation Functions

irr([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *expression*)

returns the aggregated internal rate of return for a series of cash flows represented by the numbers in *expression* iterated over the chart dimension(s). These cash flows do not have to be even, as they would be for an annuity. However, the cash flows must occur at regular intervals, such as monthly or annually. The internal rate of return is the interest rate received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods. The function needs at least one positive and one negative value to calculate. Text values, null values and missing values are disregarded. See also the **rangeirr** function (page 373).

Expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before *expression* the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **total** qualifier may be followed by a list of one or more field names within angle brackets. These field names should be a subset of the chart dimension variables. In this case the calculation will be made disregarding all chart dimension variables except those listed, i.e. one value will be returned for each combination of field values in the listed dimension fields. Also fields which are not currently a dimension in a chart may be included in the list. This may be useful in the case of group dimensions, where the dimension fields are not fixed. Listing all of the variables in the group causes the function to work when the cycle or drill-down level changes.

In previous QlikView versions, the **all** qualifier may occur before an expression. This is equivalent to using **{1} total**, i.e. in such a case, the calculation will be made over all the values of the field in the document, disregarding the chart dimensions and current selections. (The same value is always returned regardless of the logical state in the document.) For legacy reasons, **all** will still work. However, if the **all** qualifier is used, a set expression cannot be used, since the **all** qualifier defines a set by itself.

Examples:

```
irr( Payments )
irr( X*Y/3 )
irr( total Payments )
irr( total <Grp> Payments )
```

xirr ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *valueexpression*, *dateexpression*)

returns the aggregated internal rate of return for a schedule of cash flows (that is not necessarily periodic) represented by paired numbers in *valueexpression* and *dateexpression* iterated over the chart dimension(s). All payments are discounted based on a 365-day year. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair being disregarded. See also the **rangexirr** function (page 374).

Valueexpression and *dateexpression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of

records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **xirr** function supports sets and the **total** qualifier as described under the **irr** aggregation function, page 326.

Examples:

xirr(Payments, Dates)

xirr(A/B, X*Y/3)

xirr(**total** Payments, Dates)

xirr(**total** <Grp> Payments, Dates)

npv ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *rate*, *expression*)
returns the aggregated net present value of an investment based on a discount rate and a series of future payments (negative values) and incomes (positive values) represented by the numbers in *expression* iterated over the chart dimension(s). The result has a default number format of money. Rate is the interest rate per period. The payments and incomes are assumed to occur at the end of each period. Text values, null values and missing values are disregarded. See also the **rangenpv** function (page 374).

Rate and *expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **npv** function supports sets and the **total** qualifier as described under the **irr** aggregation function, page 326.

Examples:

npv(0.1, Payments)

npv(0.1, X*Y/3)

npv(**total** 0.1, Payments)

npv(**total** <Grp> 0.1, Payments)

xnpv ([*set_expression*] [**distinct**] [**total** [<fld {, fld}>]] *rate*, *valueexpression*, *dateexpression*)

returns the aggregated net present value for a schedule of cash flows (not necessarily periodic) represented by paired numbers in *valueexpression* and *dateexpression* iterated over the chart dimension(s). Rate is the interest rate per period. The result has a default number format of money. All payments are discounted based on a 365-day year. Text values, null values and missing values in any or both pieces of a data-pair will result in the entire data-pair being disregarded. See also the **rangexnpv** function (page 374).

Rate, *valueexpression* and *dateexpression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of the *expression* will be disregarded.

If the word **total** occurs before a field the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **xnpv** function supports sets and the **total** qualifier as described under the **irr** aggregation function, page 326.

Examples:

xnpv(0.1, Payments, Dates)

xnpv(0.1, A/B, X*Y/3)

xnpv(**total** 0.1, Payments, Dates)

xnpv(**total** <Grp> 0.1, Payments, Dates)

Statistical Test Functions

The following three functions apply to chi2-tests.

chi2test_p ([*set_expression*][**total** [<*fld* {*fld*>]] *col*, *row*, *observed_value* [, *expected_value*])

returns the aggregated chi2-test p value (significance) for one or two series of values iterated over the chart dimension(s). The test can be done either on the values in *observed_value* testing for variations within the specified *col* and *row* matrix or by comparing values in *observed_value* with corresponding values in *expected_values*. Text values, null values and missing values in the value expressions will result in the function returning null.

The function argument expressions must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the **aggr** function (see page 353) in combination with calculated dimensions (see page 34).

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by set analysis. Read more about “Set Analysis” on page 353.

If the word **total** occurs before the function arguments the calculation will be made over all possible values given the current selections but disregarding the chart dimension variables.

The **total** qualifier may be followed by a list of one or more field names within angle brackets. These field names should be a subset of the chart dimension variables. In this case the calculation will be made disregarding all chart dimension variables except those listed, i.e. one value will be returned for each combination of field values in the listed dimension fields. Also fields which are not currently a dimension in a chart may be included in the list. This may be useful in the case of group dimensions, where the dimension fields are not fixed. Listing all of the variables in the group causes the function to work when the cycle or drill-down level changes.

In previous QlikView versions, the **all** qualifier may occur before an expression. This is equivalent to using **{1} total**, i.e. in such a case, the calculation will be made over all the values of the field in the document, disregarding the chart dimensions and current selections. (The same value is always returned regardless of the logical state in the document.) For legacy reasons, **all** will still work. However, if

the **all** qualifier is used, a set expression cannot be used, since the **all** qualifier defines a set by itself.

Examples:

chi2test_p(Grp, Grade, Count)

chi2test_p(Gender, Description, Observed, Expected)

chi2test_df([*set_expression*][**total** [*<fld {,fld}>*]] *col, row, observed_value* [*, expected_value*]

returns the aggregated chi2-test df value (degrees of freedom) for one or two series of values iterated over the chart dimension(s). See the **chi2test_p** function, page 330, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

chi2test_df(Grp, Grade, Count)

chi2test_df(Gender, Description, Observed, Expected)

chi2test_chi2 ([*set_expression*][**total** [*<fld {,fld}>*]] *col, row, observed_value* [*, expected_value*]

returns the aggregated chi2-test value for one or two series of values iterated over the chart dimension(s). See the **chi2test_p** function, page 330, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

chi2test_chi2(Grp, Grade, Count)

chi2test_chi2(Gender, Description, Observed, Expected)

The following eight functions apply to two independent samples' student's t-tests.

TTest_t ([set_expression][**total** [<fld {,fld}>]] group, value [, eq_var = true])

returns the aggregated t value for two independent series of values iterated over the chart dimension(s). The values should be returned by *value* and be logically grouped via exactly two values in *group*. If *eq_var* is specified as false, separate variances of the two samples will be assumed, else equal variances between the samples will be assumed. Text values, null values and missing values in *value* will result in the function returning null.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about "Set Analysis" on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_t(Group, Value)

ttest_t(Group, Value, false)

TTest_df ([set_expression][**total** [<fld {,fld}>]] group, value [, eq_var = true])

returns the aggregated student's t-test df value (degrees of freedom) for two independent series of values iterated over the chart dimension(s). See the **TTest_t** function, page 332, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about "Set Analysis" on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_df(Group, Value)

ttest_df(Group, Value, false)

TTest_sig ([*set_expression*][**total** [<*fld* {*fld*>}] *group*, *value* [, *eq_var* = *true*])

returns the aggregated student's t-test 2-tailed level of significance for two independent series of values iterated over the chart dimension(s). See the **TTest_t** function, page 332, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_sig(Group, Value)

ttest_sig(Group, Value, false)

TTest_dif ([*set_expression*][**total** [<*fld* {*fld*>}] *group*, *value* [, *eq_var* = *true*])

returns the aggregated student's t-test mean difference for two independent series of values iterated over the chart dimension(s). See the **TTest_t** function, page 332, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_dif(Group, Value)

ttest_dif(Group, Value, false)

TTest_sterr ([*set_expression*][**total** [<fld {,fld}>]] *group*, *value* [, *eq_var* = *true*])

returns the aggregated student's t-test standard error of the mean difference for two independent series of values iterated over the chart dimension(s). See the **TTest_t** function, page 332, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_sterr(Group, Value)

ttest_sterr(Group, Value, false)

TTest_conf ([*set_expression*][**total** [<fld {,fld}>]] *group*, *value* [, *sig* = 0.025 [, *eq_var* = *true*]])

returns the aggregated t value for two independent series of values iterated over the chart dimension(s). The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTest_t** function, page 332, for descriptions of the other arguments.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_conf(Group, Value)

ttest_conf(Group, Value, false)

TTest_lower ([*set_expression*][**total** [*<fld {,fld}>*]] *group*, *value* [, *sig* = 0.025 [, *eq_var* = true]])

returns the aggregated value for the lower end of the confidence interval for two independent series of values iterated over the chart dimension(s). See the **TTest_t**, page 332, and **TTest_conf**, page 334, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_lower(Group, Value)

ttest_lower(Group, Value, false)

TTest_upper ([*set_expression*][**total** [*<fld {,fld}>*]] *group*, *value* [, *sig* = 0.025 [, *eq_var* = true]])

returns the aggregated value for the upper end of the confidence interval for two independent series of values iterated over the chart dimension(s). See the **TTest_t**, page 332, and **TTest_conf**, page 334, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest_upper(Group, Value)

ttest_upper(Group, Value, false)

The following eight functions apply to independent two-sample student's t-tests where the input data series is given in weighted two-column format.

TTestw_t ([*set_expression*][**total** [<*fld* {*fld*>]] *weight*, *group*, *value* [, *eq_var* = true])

returns the aggregated t value for two independent series of values iterated over the chart dimension(s). The values should be returned by *value* and be logically grouped via exactly two values in *group*. Each value in *value* can be counted one or more times according to a corresponding weight value in *weight*. If *eq_var* is specified as false, separate variances of the two samples will be assumed, else equal variances between the samples will be assumed. Text values, null values and missing values in *value* will result in the function returning null.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_t(Weight, Group, Value)

TTestw_t(Weight, Group, Value, false)

TTestw_df ([*set_expression*][**total** [<*fld* {*fld*>]] *weight*, *group*, *value* [, *eq_var* = true])

returns the aggregated student's t-test df value (degrees of freedom) for two independent series of values iterated over the chart dimension(s). See the **TTestw_t** function, page 336, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_df(Weight, Group, Value)

TTestw_df(Weight, Group, Value, false)

TTestw_sig ([*set_expression*][**total** [<*fld* {*,fld*>]] *weight*, *group*, *value* [, *eq_var* = true])

returns the aggregated student's t-test 2-tailed level of significance for two independent series of values iterated over the chart dimension(s). See the **TTestw_t** function, page 336, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_sig(Weight, Group, Value)

TTestw_sig(Weight, Group, Value, false)

TTestw_dif ([*set_expression*][**total** [<*fld* {*,fld*>]] *weight*, *group*, *value* [, *eq_var* = true])

returns the aggregated student's t-test mean difference for two independent series of values iterated over the chart dimension(s). See the **TTestw_t** function, page 336, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_dif(Weight, Group, Value)

TTestw_dif(Weight, Group, Value, false)

TTestw_sterr ([*set_expression*][**total** [<*fld* {*,fld*>]] *weight*, *group*, *value* [, *eq_var* = true])

returns the aggregated student's t-test standard error of the mean difference for two independent series of values iterated over the chart

dimension(s). See the **TTestw_t** function, page 336, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_sterr(Weight, Group, Value)

TTestw_sterr(Weight, Group, Value, false)

TTestw_conf ([set_expression][**total** [<fld {,fld}>]] weight, group, value [, sig = 0.025 [, eq_var = true]])

returns the aggregated t value for two independent series of values iterated over the chart dimension(s). The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTestw_t** function, page 336, for descriptions of the other arguments.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_conf(Weight, Group, Value)

TTestw_conf(Weight, Group, Value, false)

TTestw_lower ([set_expression][**total** [<fld {,fld}>]] weight, group, value [, sig = 0.025 [, eq_var = true]])

returns the aggregated value for the lower end of the confidence interval for two independent series of values iterated over the chart dimension(s). See the **TTestw_t**, page 336, and **TTestw_conf**, page 338, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_lower(Weight, Group, Value)

TTestw_lower(Weight, Group, Value, false)

TTestw_upper ([set_expression][**total** [<fld {,fld}>]] weight, group, value [, sig = 0.025 [, eq_var = true]])

returns the aggregated value for the upper end of the confidence interval for two independent series of values iterated over the chart dimension(s). See the **TTestw_t**, page 336, and **TTestw_conf**, page 338, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

TTestw_upper(Weight, Group, Value)

TTestw_upper(Weight, Group, Value, false)

The following eight functions apply to one-sample student's t-tests.

TTest1_t ([set_expression][**total** [<fld {,fld}>]] value)

returns the aggregated t value for a series of values iterated over the chart dimension(s). The values should be returned by *value*. Text values, null values and missing values in *value* will result in the function returning null.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of

records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:

ttest1_t(Value)

TTest1_df ([*set_expression*] [**total** [<fld {,fld}>]] value)

returns the aggregated student's t-test df value (degrees of freedom) for a series of values iterated over the chart dimension(s). See the **TTest1_t** function, page 339, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:

ttest1_df(Value)

TTest1_sig ([*set_expression*] [**total** [<fld {,fld}>]] value)

returns the aggregated student's t-test 2-tailed level of significance for a series of values iterated over the chart dimension(s). See the **TTest1_t** function, page 339, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:

ttest1_sig(Value)

TTest1_dif ([*set_expression*][**total** [<*fld* {,*fld*>}] *value*)

returns the aggregated student's t-test mean difference for a series of values iterated over the chart dimension(s). See the **TTest1_t** function, page 339, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about "Set Analysis" on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:

ttest1_dif(Value)

TTest1_sterr ([*set_expression*][**total** [<*fld* {,*fld*>}] *value*)

returns the aggregated student's t-test standard error of the mean difference for a series of values iterated over the chart dimension(s). See the **TTest1_t** function, page 339, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about "Set Analysis" on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:

ttest1_sterr(Value)

TTest1_conf ([*set_expression*][**total** [<*fld* {,*fld*>}] *value* [, *sig* = 0.025])

returns the aggregated t value for a series of values iterated over the chart dimension(s). The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTest1_t** function, page 339, for descriptions of the other arguments.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of

records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1_conf(Value)

ttest1_conf(Value, 0.005)

TTest1_lower ([*set_expression*][**total** [*<fld {,fld}>*]] *value* [, *sig = 0.025*])

returns the aggregated value for the lower end of the confidence interval for a series of values iterated over the chart dimension(s). See the **TTest1_t**, page 339, and **TTest1_conf**, page 341, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1_lower(Value)

ttest1_lower(Value, 0.005)

TTest1_upper ([*set_expression*][**total** [*<fld {,fld}>*]] *value* [, *sig = 0.025*])

returns the aggregated value for the upper end of the confidence interval for a series of values iterated over the chart dimension(s). See the **TTest1_t**, page 339, and **TTest1_conf**, page 341, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function

supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1_upper(Value)

ttest1_upper(Value, 0.005)

The following eight functions apply to one-sample student's t-tests where the input data series is given in weighted two-column format.

TTest1w_t ([*set_expression*][**total** [<*fld* {,*fld*>}] *weight*, *value*)

returns the aggregated t value for a series of values iterated over the chart dimension(s). The values should be returned by *value*. Each value in *value* can be counted one or more times according to a corresponding weight value in *weight*. Text values, null values and missing values in *value* will result in the function returning null.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_t(Weight, Value)

TTest1w_df ([*set_expression*][**total** [<*fld* {,*fld*>}] *weight*, *value*)

returns the aggregated student's t-test df value (degrees of freedom) for a series of values iterated over the chart dimension(s). See the **TTest1w_t** function, page 343, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_df(Weight, Value)

TTest1w_sig ([*set_expression*][**total** [<*fld* {*fld*>]] *weight, value*)
returns the aggregated student's t-test 2-tailed level of significance for a series of values iterated over the chart dimension(s). See the **TTest1w_t** function, page 343, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_sig(Weight, Value)

TTest1w_dif ([*set_expression*][**total** [<*fld* {*fld*>]] *weight, value*)
returns the aggregated student's t-test mean difference for a series of values iterated over the chart dimension(s). See the **TTest1w_t** function, page 343, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_dif(Weight, Value)

TTest1w_sterr ([*set_expression*][**total** [<*fld* {*fld*>]] *weight, value*)
returns the aggregated student's t-test standard error of the mean difference for a series of values iterated over the chart dimension(s). See the **TTest1w_t** function, page 343, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_sterr(Weight, Value)

TTest1w_conf ([set_expression][**total** [<fld {,fld}>]] weight, value [, sig = 0.025])

returns the aggregated t value for a series of values iterated over the chart dimension(s). The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **TTest1w_t** function, page 343, for descriptions of the other arguments.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_conf(Weight, Value)

ttest1w_conf(Weight, Value, 0.005)

TTest1w_lower ([set_expression][**total** [<fld {,fld}>]] weight, value [, sig = 0.025])

returns the aggregated value for the lower end of the confidence interval for a series of values iterated over the chart dimension(s). See the **TTest1w_t**, page 343, and **TTest1w_conf** , page 345, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_lower(Weight, Value)

ttest1w_lower(Weight, Value, 0.005)

TTest1w_upper ([*set_expression*][**total** [<*fld* {*fld*>]] *weight*, *value* [, *sig* = 0.025])

returns the aggregated value for the upper end of the confidence interval for a series of values iterated over the chart dimension(s). See the **TTest1w_t**, page 343, and **TTest1w_conf** , page 345, functions for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ttest1w_upper(Weight, Value)

ttest1w_upper(Weight, Value, 0.005)

The following five functions apply to z-tests.

ZTest_z ([*set_expression*][**total** [<*fld* {*fld*>]] *value* [, *sigma*])

returns the aggregated z value for a series of values iterated over the chart dimension(s). The values should be returned by *value*. A population mean of 0 is assumed. If you want the test to be performed around another mean, subtract that value from the sample values. If known, the standard deviation can be stated in *sigma*. If *sigma* is omitted the actual sample standard deviation will be used. Text values, null values and missing values in *value* will result in the function returning null.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:**ztest_z**(Value-TestValue)**ZTest_sig** ([*set_expression*][**total** [*<fld {,fld}>*]] *value* [, *sigma*])

returns the aggregated z-test 2-tailed level of significance for a series of values iterated over the chart dimension(s). See the **ZTest_z** function, page 346, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:**ztest_sig**(Value-TestValue)**ZTest_dif** ([*set_expression*][**total** [*<fld {,fld}>*]] *value* [, *sigma*])

returns the aggregated z-test mean difference for a series of values iterated over the chart dimension(s). See the **ZTest_z** function, page 346, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:**ztest_dif**(Value-TestValue)**ZTest_sterr** ([*set_expression*][**total** [*<fld {,fld}>*]] *value* [, *sigma*])

returns the aggregated z-test standard error of the mean difference for a series of values iterated over the chart dimension(s). See the **ZTest_z** function, page 346, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of

records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Example:

ztest_sterr(Value-TestValue)

ZTest_conf ([*set_expression*][**total** [<*fld* {*fld*>]] *value* [, *sigma* [, *sig* = 0.025]])

returns the aggregated z value for a series of values iterated over the chart dimension(s). The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **ZTest_z** function, page 346, for descriptions of the other arguments.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ztest_conf(Value-TestValue)

The following five functions apply to z-tests where the input data series is given in weighted two-column format.

ZTestw_z ([*set_expression*][**total** [<*fld* {*fld*>]] *weight*, *value* [, *sigma*])

returns the aggregated z value for a series of values iterated over the chart dimension(s). The values should be returned by *value*. A sample mean of 0 is assumed. If you want the test to be performed around another mean, subtract that value from the sample values. Each value in *value* can be counted one or more times according to a corresponding weight value in *weight*. If known, the standard deviation can be stated in *sigma*. If *sigma* is omitted the actual sample standard deviation will be used. Text values, null values and missing values in *value* will result in the function returning null.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ztestw_z(Weight, Value-TestValue)

ZTestw_sig ([*set_expression*][**total** [*<fld {,fld}>*]] *weight, value* [, *sigma*]
)

returns the aggregated z-test 2-tailed level of significance for a series of values iterated over the chart dimension(s). See the **ZTestw_z** function, page 348, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ztestw_sig(Weight, Value-TestValue)

ZTestw_dif ([*set_expression*][**total** [*<fld {,fld}>*]] *weight, value* [, *sigma*]
)

returns the aggregated z-test mean difference for a series of values iterated over the chart dimension(s). See the **ZTestw_z** function, page 348, function for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ztestw_dif(Weight, Value-TestValue)

ZTestw_sterr ([*set_expression*][**total** [<*fld* {*fld*>]] *weight*, *value* [, *sigma*])

returns the aggregated z-test standard error of the mean difference for a series of values iterated over the chart dimension(s). See the **ZTestw_z** function, page 348, for argument descriptions.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ztestw_sterr(Weight, Value-TestValue)

ZTestw_conf ([*set_expression*][**total** [<*fld* {*fld*>]] *weight*, *value* [, *sigma*] [, *sig* = 0.025])

returns the aggregated z value for a series of values iterated over the chart dimension(s). The two-tailed level of significance can be specified in *sig*. If omitted *sig* will be set to 0.025, resulting in a 95% confidence interval. See the **ZTestw_z** function, page 348, for descriptions of the other arguments.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The same limitations with regard to nested aggregation as those listed under the **chi2test_p** function, page 330, apply. This function supports sets and the **total** qualifier in the same manner as described under the **chi2test_p** function, page 330.

Examples:

ztestw_conf(Weight, Value-TestValue)

Special Input Field Aggregation Function

inputsum (*inputfield* [, *distribution_mode*][*set_expression*])

Returns the aggregated sum of *inputfield* iterated over the chart dimension(s). *Inputfield* must be a field name of a field properly declared as an input field in the script.

When this aggregation function is used as expression in a table chart, it will be possible to edit the aggregated sum interactively. When hovering over the expression cell you will see an input icon. Clicking the icon sets the cell in input edit mode. It is possible to use up/down arrow keys to move between cells while staying in input edit mode. The change in the aggregated sum will be distributed to the underlying field values using the chosen *distribution_mode*. The entire QlikView document will automatically recalculate whenever new values are entered.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The *distribution_mode* parameter can have the following values:

- '+' 'Default mode. Equal parts of the change is distributed to all the underlying values.
- '*' 'The change is distributed proportionally (to existing values) to the underlying values.
- '=' 'The value entered is given to all the underlying values.
- '/' 'The value is divided equally between underlying values, without regard to previous distribution but keeping previous sum.

The *distribution_mode* values above can be amended with the following modifiers:

- T E.g. '+T'. Will cause a compensation of the entered change over the other selected values in the input field (keeping the grand total intact).
- A E.g. '+A'. Will cause a compensation of the entered change over all other values (keeping the grand total intact).

Examples:

```
inputsum( Budget )
inputsum( Budget, '+' )
inputsum( Budget, '*' )
inputsum( Budget, '=' )
```

inputsum(Budget, '/')
inputsum(Budget, '+T')
inputsum(Budget, '+A')

inputavg (*inputfield* [, *distribution_mode*][*set_expression*])

Returns the aggregated average of *inputfield* iterated over the chart dimension(s). *Inputfield* must be a field name of a field properly declared as an input field in the script.

When this aggregation function is used as expression in a table chart, it will be possible to edit the aggregated value interactively. When hovering over the expression cell you will see an input icon. Clicking the icon sets the cell in input edit mode. It is possible to use up/down arrow keys to move between cells while staying in input edit mode. The change in the aggregated value will be distributed to the underlying field values using the chosen *distribution_mode*. The entire QlikView document will automatically recalculate whenever new values are entered.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

The *distribution_mode* parameter can have the following values:

- '+' 'Default mode. Equal parts of the change is distributed to all the underlying values.
- '*' 'The change is distributed proportionally (to existing values) to the underlying values.
- '=' 'The value entered is given to all the underlying values.
- '/' 'The value is divided equally between underlying values, without regard to previous distribution but keeping previous average.

The *distribution_mode* values above can be amended with the following modifiers:

- T E.g. '+T'. Will cause a compensation of the entered change over the other selected values in the input field (keeping the grand total intact).
- A E.g. '+A'. Will cause a compensation of the entered change over all other values (keeping the grand total intact).

Examples:**inputavg**(Budget)**inputavg** (Budget, '+')**inputavg** (Budget, '*')**inputavg** (Budget, '=')**inputavg** (Budget, '/')**inputavg** (Budget, '+T')**inputavg** (Budget, '+A')

Special Function for Advanced Aggregations

aggr ([**distinct** / **nodistinct**] [*set_expression*] *expression* {, *dimension*})
 returns a set of values of *expression* calculated over *dimensions*. The result can be compared to the expression column of a 'local chart', evaluated in the context where the **aggr** function resides. Each *dimension* must be a single field. It may not be an expression (calculated dimension).

If the *expression* argument is preceded by the **nodistinct** qualifier, each combination of dimension values may generate more than one return value, depending on underlying data structure. If the *expression* argument is preceded by the **distinct** qualifier or if no qualifier is used at all, each combination of dimension values will generate only one return value.

By default, the aggregation function will aggregate over the set of possible records defined by the selection. An alternative set of records can be defined by a set expression. Read more about “Set Analysis” on page 353.

By using this function in calculated dimensions (see page 34) it is possible to achieve nested chart aggregation.

When used in chart expressions it is possible to achieve sum of rows totals in a pivot table (see page 461).

Examples:**aggr**(sum(Sales), Country)**aggr**(**nodistinct** sum(Sales), Country)**aggr**(sum(Sales), Country, Region)**count**(**aggr**(sum(Sales), Country))

Set Analysis

Sets can be used in aggregation functions. Aggregation functions normally aggregate over the set of possible records defined by the current selection.

But an alternative set of records can be defined by a set expression. Hence, a set is conceptually similar to a selection.

A set expression always begins and ends with curly brackets when used, e.g. **{BM01}**.

Set Identifiers

There is a constant that can be used to denote a record set; **1**. It represents the full set of all the records in the application.

The **\$** sign represents the records of the current selection. The set expression **{\$}** is thus the equivalent of not stating a set expression. **{1-\$}** defines the inverse of the current selection, i.e. everything that the current selections excludes.

Selections from the Back/Forward stack can be used as set identifiers, by use of the dollar symbol: **\$1** represents the previous selection, i.e. equivalent to pressing the **Back** button. Similarly, **\$_1** represents one step forward, i.e. equivalent to pressing the **Forward** button. Any unsigned integer can be used in the **Back** and **Forward** notations, i.e. **\$0** represents the current selection.

Finally, bookmarks can be used as set identifiers. Note that only server and document bookmarks can be used as set identifiers. Either the bookmark ID or the bookmark name can be used, e.g. **BM01** or **MyBookmark**. Only the selection part of a bookmark is used. Variable values are not included. It is thus not possible to use input fields in bookmarks for set analysis.

Examples:

sum({\$} Sales)

returns sales for the current selection, i.e. the same as **sum(Sales)**.

sum({\$1} Sales)

returns sales for the previous selection.

sum({\$_2} Sales)

returns sales for the 2nd next selection, i.e. two steps forward. Only relevant if you just made two **Back** operations.

sum({1} Sales)

returns the total sales within the application, disregarding the selection but not the dimension. If used in a chart with e.g. Products as dimension, each product will get a different value.

sum({1} Total Sales)

returns total sales within the application, disregarding both selection and dimension. I.e. the same as **sum(All Sales)**.

sum({BM01} Sales)

returns sales for the bookmark BM01.

sum({MyBookMark} Sales)

returns sales for the bookmark MyBookmark.

sum({Server\BM01} Sales)

returns the sales for the server bookmark BM01.

sum({Document\MyBookmark}Sales)

returns the sales for the document bookmark MyBookmark.

Set Operators

Several set operators that can be used in set expressions exist. All set operators use sets as operands, as described above, and return a set as result.

+ Union. This binary operation returns a set consisting of the records that belong to any of the two set operands.

- Exclusion. This binary operation returns a set of the records that belong only to the first of the two set operands. Also, when used as a unary operator, it returns the complement set.

***** Intersection. This binary operation returns a set consisting of the records that belong to both of the two set operands.

/ Symmetric difference (XOR). This binary operation returns a set consisting of the records that belong to either, but not both of the two set operands.

The order of precedence is 1) Unary minus (complement), 2) Intersection and Symmetric difference, and 3) Union and Exclusion.

Within a group, the expression is evaluated from left to right. Alternative orders can be defined by standard brackets, which may be necessary since the set operators do not commute, e.g. $A+(B-C)$ is different from $(A+B)-C$ which in turn is different from $(A-C)+B$.

Examples:

sum({1-\$} Sales)

returns sales for everything excluded by the current selection.

sum({\$*BM01} Sales)

returns sales for the intersection between the current selection and bookmark BM01.

sum({-(\$+BM01)} Sales)

returns sales excluded by current selection and bookmark BM01.

Note The use of set operators in combination with basic aggregation expressions involving fields from multiple QlikView tables may cause unpredictable results and should be avoided. E.g. if *Quantity* and *Price* are fields from different tables, then the expression **sum**({ \$*BM01 }Quantity*Price) should be avoided.

Set Modifiers

A set can be modified by an additional or a changed selection. Such a modification can be written in the set expression. The modifier consists of one or several field names, each followed by a selection that should be made on the field, all enclosed by < and >. E.g. <Year={2007,2008},Region={US}>. Field names and field values can, be quoted as usual, e.g. <[Sales Region]={'West coast', 'South America'}>.

There are several ways to define the selection: A simple case is a selection based on the selected values of another field, e.g. <OrderDate = DeliveryDate>. This modifier will take the selected values from *DeliveryDate* and apply those as a selection on *OrderDate*. If there are many distinct values – more than a couple of hundred – then this operation is CPU intense and should be avoided.

The most common case, however, is a selection based on a field value list enclosed in curly brackets, the values separated by commas, e.g. <Year = {2007, 2008}>. The curly brackets here define an element set, where the elements can be either field values or searches of field values. A search is always defined by the use of double quotes, e.g. <Ingredient = {"*Garlic*"}> will select all ingredients including the string 'garlic'. Searches are case-insensitive and are made also over excluded values.

Empty element sets, either explicitly e.g. <Product = {}> or implicitly e.g. <Product = {"Perpetuum Mobile"}> (a search with no hits) mean no product, i.e. they will result in a set of records that are not associated with any product. Note that this set cannot be achieved through usual selections, unless a selection is made in another field, e.g. *TransactionID*.

Finally, for fields in **and**-mode, there is also the possibility of forced exclusion. If you want to force exclusion of specific field values, you will need to use “~” in front of the field name.

A set modifier can be used on a set identifier or on its own. It cannot be used on a set expression. When used on a set identifier, the mod-

ifier must be written immediately after the set identifier, e.g. `{<Year = {2007, 2008}>}`. When used on its own, it is interpreted as a modification of the current selection.

Examples:

sum({1<Region= {US} >} Sales)

returns the sales for the region US disregarding the current selection.

sum({<Region = >} Sales)

returns the sales for the current selection, but with the selection in *Region* removed.

sum({<Region = >} Sales)

returns the same as the example immediately above. When the set to modify is omitted, \$ is assumed.

Note The syntax in the two previous examples is interpreted as no selections in *Region*, i.e. all regions given other selections will be possible. It is not equivalent to the syntax `<Region = { }>` (or any other text on the right side of the equal sign implicitly resulting in an empty element set) which is interpreted as no region.

sum({<Year = {2000}, Region = {US, SE, DE, UK, FR}>} Sales)

returns the sales for the current selection, but with new selections both in *Year* and in *Region*.

sum({<~Ingredient = {"*garlic*"}>} Sales)

returns the sales for the current selection, but with a forced exclusion of all *Ingredients* containing the string 'garlic'.

sum({<Year = {"2*"}>} Sales)

returns the sales for the current selection, but with all years beginning with the digit 2, i.e. most likely year 2000 and onwards, selected in the field *Year*.

sum({<Year = {"2*", "198*"}>} Sales)

as above, but now the 1980:s are also included in the selection.

sum({<Year = {">1978<2004"}>} Sales)

as above, but now with a numeric search so that an arbitrary range can be specified.

Set Modifiers with Set Operators

The selection within a field can be defined using set operators as described above, working on different element sets, e.g. the modifier $\langle \text{Year} = \{ "20*", 1997 \} - \{ 2000 \} \rangle$ will select all years beginning with “20” in addition to “1997”, except for “2000”.

Examples

$\text{sum}(\{ \$\langle \text{Product} = \text{Product} + \{ \text{OurProduct1} \} - \{ \text{OurProduct2} \} \rangle \} \text{Sales})$

returns the sales for the current selection, but with the product *OurProduct1* added to the list of selected products and *OurProduct2* removed from the list of selected products.

$\text{sum}(\{ \$\langle \text{Year} = \text{Year} + (\{ "20*", 1997 \} - \{ 2000 \}) \rangle \} \text{Sales})$

returns the sales for the current selection but with additional selections in the field *Year*: 1997 and all that begin with 20 – however, not 2000. Note that if 2000 is included in the current selection, it will still be included after the modification.

$\text{sum}(\{ \$\langle \text{Year} = (\text{Year} + \{ "20*", 1997 \}) - \{ 2000 \} \rangle \} \text{Sales})$

returns almost the same as above, but here 2000 will be excluded, even if it initially is included in the current selection. The example shows the importance of sometimes using brackets to define an order of precedence.

$\text{sum}(\{ \$\langle \text{Year} = \{ "*" \} - \{ 2000 \}, \text{Product} = \{ "*" \text{bearing} "*" \} \rangle \} \text{Sales})$

returns the sales for the current selection but with a new selection in *Year*: all years except 2000; and only for products containing the string ‘bearing’.

Set Modifiers Using Assignments with Implicit Set Operators

The above notation defines new selections, disregarding the current selection in the field. However, if you want to base your selection on the current selection in the field and add field values, you may for example want a modifier $\langle \text{Year} = \text{Year} + \{ 2007, 2008 \} \rangle$. A short and equivalent way to write this is $\langle \text{Year} += \{ 2007, 2008 \} \rangle$, i.e. the assignment operator implicitly defines a union. Implicit intersections, exclusions and symmetric differences can also be defined using $\langle "*" = \rangle$, $\langle "*" = \rangle$ and $\langle "/= \rangle$.

Examples:

sum({<Product += {OurProduct1, OurProduct2} >} Sales)

returns the sales for the current selection, but using an implicit union to add the products *OurProduct1* and *OurProduct2* to the list of selected products.

sum({<Year += {"20*",1997} - {2000} >} Sales)
 returns the sales for the current selection but using an implicit union to add a number of years in the selection: 1997 and all that begin with 20 – however, not 2000. Note that if 2000 is included in the current selection, it will still be included after the modification. Same as <Year=Year + ({“20*”,1997}–{2000})>

sum({<Product *= {OurProduct1} >} Sales)
 returns the sales for the current selection, but only for the intersection of currently selected products and the product *OurProduct1*.

Set Modifiers with Dollar-sign expansions

Variables and other dollar-sign expansions can be used in set expressions.

Examples:

sum({<Year = {\$(#vLastYear)}>} Sales)
 returns the sales for the previous year in relation to the current selection. Here, a variable *vLastYear* containing the relevant year is used in a dollar-sign expansion.

sum({<Year = {\$(#=Only(Year)-1)}>} Sales)
 returns the sales for the previous year in relation to the current selection. Here, a dollar-sign expansion is used to calculate previous year.

Set Modifiers with Advanced Searches

Advanced searches using wildcards and aggregations can be used to define sets.

Examples:

sum({<Product = {"*Internal*", "*Domestic*"}>} Sales)

returns the sales for the current selection, excluding transactions pertaining to products with the string ‘Internal’ or ‘Domestic’ in the product name.

sum({ \$<Customer = {“=Sum({1<Year = {2007}>} Sales) > 1000000”}>} Sales)

returns the sales for the current selection, but with a new selection in the *Customer* field: only customers who during 2007 had a total sales of more than 1,000,000.

Set Modifiers with Implicit Field Value Definitions

In the above examples, all field values have been explicitly defined or defined through searches. There is however an additional way to define a set of field values by the use of a nested set definition.

In such cases, the element functions P() and E() must be used, representing the element set of possible values and the excluded values of a field, respectively. Inside the brackets, it is possible to specify one set expression and one field, e.g. P({1} Customer). These functions cannot be used in other expressions:

Examples:

sum({ \$<Customer = P({1<Product={‘Shoe’}>} Customer)>} Sales)

returns the sales for current selection, but only those customers that ever have bought the product ‘Shoe’. The element function P() here returns a list of possible customers; those that are implied by the selection ‘Shoe’ in the field Product.

sum({ \$<Customer = P({1<Product={‘Shoe’}>})>} Sales)

same as above. If the field in the element function is omitted, the function will return the possible values of the field specified in the outer assignment.

sum({ \$<Customer = P({1<Product={‘Shoe’}>} Supplier)>} Sales)

returns the sales for current selection, but only those customers that ever have supplied the product ‘Shoe’. The element function P() here returns a list of possible suppliers; those that are implied by the selection ‘Shoe’ in the field Product. The list of suppliers is then used as a selection in the field Customer.

sum({ \$<Customer = E({1<Product={‘Shoe’}>})>} Sales)

returns the sales for current selection, but only those customers that never bought the product ‘Shoe’. The element function E() here returns the list of excluded customers;

those that are excluded by the selection 'Shoe' in the field Product.

Syntax for Sets

Hence, the full syntax (not including the optional use of standard brackets to define precedence) is

set_expression ::= { *set_entity* { *set_operator* *set_entity* } }

set_entity ::= *set_identifier* [*set_modifier*]

set_identifier ::= **1** | \$ | \$N | \$_N | *bookmark_id* | *bookmark_name*

set_operator ::= + | - | * | /

set_modifier ::= < *field_selection* { , *field_selection* } >

field_selection ::= *field_name* [= | += | -= | *= | /=]

element_set_expression

element_set_expression ::= *element_set* { *set_operator* *element_set* }

element_set ::= [*field_name*] / { *element_list* } | *element_function*

element_list ::= *element* { , *element* }

element_function ::= (**P** | **E**) ([*set_expression*] [*field_name*])

element ::= *field_value* | " *search_mask* "

19.5 Synthetic Dimension Functions

ValueList (*value* { , *value* })

returns a set of listed values which, when used in a calculated dimension, will form a synthetic dimension. In charts with a synthetic dimension created with the **valuelist** function it is possible to reference the dimension value corresponding to a specific expression cell by restating the **valuelist** function with the same parameters in the chart expression. The function may of course be used anywhere in the layout, but apart from when used for synthetic dimensions it will only be meaningful inside an aggregation function.

Examples:

valuelist (1, 10, 100)

valuelist ('a', 'xyz', 55)

ValueLoop(from [, to [, step = 1]])

returns a set of iterated values which, when used in a calculated dimension, will form a synthetic dimension. The values generated will start with the from value and end with the to value including intermediate values in increments of step. In charts with a synthetic dimension created with the **valueloop** function it is possible to reference the dimension value corresponding to a specific expression cell by restating the **valueloop** function with the same parameters in the chart expression. The function may of course be used anywhere in the layout, but apart from when used for synthetic dimensions it will only be meaningful inside an aggregation function.

Examples:

valueloop (1, 3) returns the values 1, 2 and 3

valueloop (1, 5, 2) returns the values 1, 3 and 5

valueloop (11) returns the value 11

19.6 Other Functions

Functions can have different numbers of parameters as well as different types of parameters. In the functions below, the parameters are expressions where x should be interpreted as a real valued number, n as an integer and s as a string.

QlikView chart expressions support close to one hundred standard functions, apart from the aggregation functions described above. The functions have been divided into groups as follows:

General numeric functions	page 363
Range functions	page 367
Exponential and logarithmic functions.....	page 375
Trigonometric and hyperbolic functions	page 375
Statistical distribution functions	page 376
Financial functions	page 379
Mathematical constants and parameter-free functions	page 383
String functions	page 383
Inter-record functions	page 387
Document functions	page 402
Ranking functions.....	page 403
Conditional functions	page 406

Logical functions..... page 408
 Null functions page 409
 System functions page 409
 Date and time functions..... page 412
 Number interpretation functions page 434
 Formatting functions page 439
 Color functions page 443

General Numeric Functions

div(*x1* , *x2*)

Integer division. Both parameters are interpreted as real numbers, i.e. they do not have to be integers. The result is the integer part of the result of the arithmetic division, a real number.

Examples:

div(7,2) returns 3

div(9,3) returns 3

div(-4,3) returns -1

div(4,-3) returns -1

div(-4,-3) returns 1

mod(*x1* , *x2*)

Mathematical modula function. Both parameters must have integer values. *x2* must be greater than 0. The result is the mathematical modula function, i.e. the non-negative remainder of an integer division.

Examples:

mod(7,2) returns 1

mod(9,3) returns 0

mod(-4,3) returns 2

mod(4,-3) returns NULL

mod(-4,-3) returns NULL

fmod(*x1* , *x2*)

Generalized modula function. Both parameters are interpreted as real numbers, i.e. they do not have to be integers. The result is the remainder of the integer division, a real number.

Examples:

fmod(7,2) returns 1

fmod(9,3) returns 0

fmod(-4,3) returns -1
fmod(4,-3) returns 1
fmod(-4,-3) returns -1

ceil(*x* [, *base* [, *offset*]])

Rounding of *x* upwards to the nearest multiple of *base* with an offset of *offset*. The result is a number.

Examples:

ceil(2.4) returns 3
ceil(2.6) returns 3
ceil(3.88 , 0.1) returns 3.9
ceil(3.88 , 5) returns 5
ceil(1.1 , 1 , 0.5) returns 1.5

floor(*x* [, *base* [, *offset*]])

Rounding of *x* downwards to the nearest multiple of *base* with an offset of *offset*. The result is a number.

Examples:

floor(2.4) returns 2
floor(2.6) returns 2
floor(3.88 , 0.1) returns 3.8
floor(3.88 , 5) returns 0
floor(1.1 , 1 , 0.5) returns 0.5

frac(*x*)

Returns the fraction part of *x*. The fraction is defined in such a way that **frac**(*x*) + **floor**(*x*) = *x*.

Examples:

frac(11.43) returns **0.43**
frac(-1.4) returns **0.6**

round(*x* [, *step* [, *offset*]])

Rounding of *x* upwards or downwards *n* number of steps with an offset of *offset*. The result is a number. If *x* is exactly in the middle of an interval, it is rounded upwards. The function complies with the IEEE standard 64-bit floating point numbers.

Examples:

round(2.4) returns 2
round(2.6) returns 3
round(2.5) returns 3
round(3.88 , 0.1) returns 3.9

round(3.88 , 5) returns 5

round(1.1 , 1 , 0.5) returns 1.5

fabs(*x*)

The absolute value of *x*. The result is a positive number.

Examples:

fabs(2.4) returns 2.4

fabs(-3.8) returns 3.8

sign(*x*)

Returns 1, 0, or -1 depending on whether *x* is a positive number, 0, or a negative number. If no numeric value is found, NULL is returned.

Examples:

sign(66) returns 1

sign(0) returns 0

sign(-234) returns -1

numsum(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangesum**.

numcount(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangenumericcount**.

numavg(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangeavg**.

nummin(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangemin**.

nummax(*expr1* [, *expr2*, ... *exprN*])

Obsolete. Superseded by range function **rangemax**.

fact(*n*)

Returns the factorial of a positive integer *n*. If the number *n* is not an integer it will be truncated. Non-positive numbers will return NULL.

Examples:

fact(1) returns 1

fact(5) returns 120 (1*2*3*4*5 = 120)

fact(-5) returns NULL

combin(*n1*, *n2*)

Returns the number of combinations of *n2* items that can be picked from a group of *n1* items. The order in which the items are picked is not significant. Non-integer arguments will be truncated.

Example:

How many combinations of seven numbers can be picked from a total of 35 lotto numbers?

combin(35,7) returns 6 724 520

permut(*n1*, *n2*)

Returns the number of permutations of *n2* items that can be selected from a group of *n1* items. The order in which the items are picked is significant. Non-integer arguments will be truncated.

Example:

In how many ways could the gold, silver and bronze medals be distributed after a 100m final with 8 participants?

permut(8,3) returns 336

even(*n*)

Returns true if *n* is an even integer, false if *n* is an odd integer and NULL if *n* is not an integer.

Examples:

even(3) returns false

even(2*10) returns true

even(3.14) returns NULL

odd(*n*)

Returns true if *n* is an odd integer, false if *n* is an even integer and NULL if *n* is not an integer.

Examples:

odd(3) returns true

odd(2*10) returns false

odd(3.14) returns NULL

bitcount(*i*)

Returns the number of set bits in *i*. *I* will be interpreted as a signed 64-bit integer, though only the first 48 bits are significant. A value above 48-bits will return NULL.

Examples:

bitcount(3) returns 2

bitcount(100) returns 3

bitcount(n)+bitcount(bitnot n) returns 64 (as long as n is a valid integer)

Range Functions

rangesum(expr1 [, expr2, ... exprN])

Returns the sum of a range of 1 to N arguments. As opposed to the + operator, **rangesum** will treat all non-numeric values as 0.

Examples:

rangesum(1,2,4) returns 7

rangesum(1,'xyz') returns 1

rangesum(null()) returns 0

rangesum (above(count(x),-1,-3)) returns the sum of the three results of the count(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangeavg(expr1 [, expr2, ... exprN])

Returns the average of a range of 1 to N arguments. If no numeric value is found, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangeavg(1,2,4) 2.33333333

rangeavg(1,'xyz') returns 1

rangeavg(null(), 'abc') returns NULL

rangeavg (above(count(x),-1,-3)) returns a sliding average of the result of the inner count(x) expression calculated the current row and two rows above the current row.

rangecount(expr1 [, expr2, ... exprN])

Returns the number of values found within a range of 1 to N arguments. Null values are not counted.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangecount (1,2,4) returns 3

rangecount (2,'xyz') returns 2

rangecount (null()) returns 0

rangecount (above(sum(x),-1,3)) returns the number of values within the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangemin(*expr1* [, *expr2*, ... *exprN*])

Returns the lowest numeric value found within a range 1 to N arguments. If no numeric value is found, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangemin (1,2,4) returns 1

rangemin (1,'xyz') returns 1

rangemin (null(), 'abc') returns NULL

rangemin (min(sum(x),-1,3)) returns the lowest of the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangemax(*expr1* [, *expr2*, ... *exprN*])

Returns the highest numeric values found within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangemax (1,2,4) returns 4

rangemax (1,'xyz') returns 1

rangemax (null(), 'abc') returns NULL

rangemax (max(sum(x),-1,3)) returns the highest of the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangestdev(*expr1* [, *expr2*, ... *exprN*])

Returns the standard deviation within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangestdev** (1,2,4) returns 1.5275252316519**rangestdev** (null()) returns NULL**rangestdev (above(count(x),0,3))** returns a sliding standard deviation of the result of the inner count(x) expression calculated on the current row and two rows above the current row.**rangeskew**(*expr1* [, *expr2*, ... *exprN*])

Returns the skewness within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangeskew** (1,2,4) returns 0.93521952958283**rangeskew (above(count(x),0,3))** returns a sliding skewness of the result of the inner count(x) expression calculated on the current row and two rows above the current row.**rangekurtosis**(*expr1* [, *expr2*, ... *exprN*])

Returns the kurtosis within a range of 1 to N arguments. If no numeric value is found, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangekurtosis** (1,2,4,7) returns -0.28571428571429**rangekurtosis (above(count(x),0,3))** returns a sliding kurtosis of the result of the inner count(x) expression calculated on the current row and two rows above the current row.**rangefractile**(*fractile*, *expr1* [, *expr2*, ... *exprN*])

Returns the number of numeric values found within a range of 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangefractile** (0.25,1,2,4,6) returns 1.5

rangefractile (0.5, above(sum(x),-1,3)) returns the median of the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangenumericcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of numeric values found within a range of 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangenumericcount (1,2,4) returns 3

rangenumericcount (2,'xyz') returns 1

rangenumericcount (null()) returns 0

rangenumericcount (above(sum(x),-1,3)) returns the number of numeric values within the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangetextcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of text values found within a range of 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangetextcount (1,2,4) returns 0

rangetextcount (2,'xyz') returns 1

rangetextcount (null()) returns 0

rangetextcount (above(sum(x),-1,3)) returns the number of text values within the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangenullcount(*expr1* [, *expr2*, ... *exprN*])

Returns the number of numeric values found within a range of 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangenullcount** (1,2,4) returns 0**rangenullcount** (2,'xyz') returns 0**rangenullcount** (null(),null()) returns 2**rangenullcount** (**above(sum(x),-1,3)**) returns the number of null values within the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.**rangemissingcount**(*expr1* [, *expr2*, ... *exprN*])

Returns the number of non-numeric values (including null values) found among 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangemissingcount** (1,2,4) returns 0**rangemissingcount** (2,'xyz') returns 1**rangemissingcount** (null()) returns 1**rangemissingcount** (**above(sum(x),-1,3)**) returns the number of non-numeric values within the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.**rangemode**(*expr1* [, *expr2*, ... *exprN*])

Returns the mode value, i.e. the most commonly occurring value within a range of 1 to N arguments. If more than one value shares the highest frequency, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:**rangemode** (1,2,9,2,4) returns 2**rangemode** ('a',4,'a',4) returns NULL**rangemode** (null()) returns NULL**rangemode** (**above(sum(x),-1,3)**) returns the most commonly occurring value within the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangeonly(*expr1* [, *expr2*, ... *exprN*])

If exactly one non-null value exists among the range of N expressions, that value will be returned. In all other cases, NULL is returned.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangeonly (1,2,4) returns NULL

rangeonly (1,'xyz') returns NULL

rangeonly (null(), 'abc') returns 'abc'

rangeonly (above(sum(x),-1,3)) uses the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row. If they contain exactly one value, that value will be returned.

rangeminstring(*expr1* [, *expr2*, ... *exprN*])

Returns the first text value in text sort order found among 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangeminstring (1,2,4) returns 1

rangeminstring ('xyz','abc') returns 'abc'

rangeminstring (null()) returns NULL

rangeminstring (above(sum(x),-1,3)) returns the first (in text sort order) of the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangemaxstring(*expr1* [, *expr2*, ... *exprN*])

Returns the last text value in text sort order found among 1 to N arguments.

The argument expressions of this function may contain inter-record functions with a third optional parameter, which in themselves return a range of values.

Examples:

rangemaxstring (1,2,4) returns 4

rangemaxstring ('xyz','abc') returns 'xyz'

rangemaxstring (null()) returns NULL

rangemaxstring (above(sum(x),-1,3)) returns the last (in text sort order) of the three results of the sum(x) function evaluated on the row below the current row, the current row and the row above the current row.

rangeccorrel(*x-value*, *y-value* {*x-value*, *y-value*})

returns correlation coefficient for a series of coordinates.

X-value is a single value or a range of values as returned by an inter-record function with a third optional parameter (page 396).

Y-value is a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 388). Each value or range of values must correspond to an *x-value* or a range of *x-values*.

The function needs at least two pairs of coordinates to calculate. Text values, null values and missing values are disregarded.

Example:

rangeccorrel(2, 3, 6, 8, 9, 4) returns 0.269

rangeirr(*value* {*value*})

returns the internal rate of return for a series of cash flows represented by the numbers in values. These cash flows do not have to be even, as they would be for an annuity. However, the cash flows must occur at regular intervals, such as monthly or annually. The internal rate of return is the interest rate received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods.

Value is a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 388). The function needs at least one positive and one negative value to calculate. Text values, null values and missing values are disregarded.

Examples:

rangeirr(-70000,12000,15000,18000,21000,26000) returns 0,0866

rangeirr(above(sum(value), 0, 10))

rangeirr(above(total value, 0, rowno(total)))

rangenpv (*rate*, *value* {,*value*})

returns the net present value of an investment based on a discount rate and a series of future payments (negative values) and incomes (positive values). The result has a default number format of money.

Rate is the interest rate per period.

Value is a payment or income occurring at the end of each period. Each *value* may be a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 388). Text values, null values and missing values are disregarded.

Examples:

rangenpv(0.1,-10000,3000,4200,6800) returns 1188,44

rangenpv(0.05, **above**(**sum**(*value*), 0, 10))

rangenpv(0.05, **above**(**total** *value*, 0, **rowno**(**total**)))

rangexirr(*value*, *date* {,*value*, *date*})

returns the internal rate of return for a schedule of cash flows that is not necessarily periodic. To calculate the internal rate of return for a series of periodic cash flows, use the **rangeirr** function.

Value is a cash flow or a series of cash flows that corresponds to a schedule of payments in dates. Each *value* may be a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 388). Text values, null values and missing values are disregarded. All payments are discounted based on a 365-day year. The series of values must contain at least one positive and one negative value.

Date is a payment date or a schedule of payment dates that corresponds to the cash flow payments.

Examples:

rangexirr(-2500,'2008-01-01',2750,'2008-09-01') returns 0,1532

rangexirr (**above**(**sum**(*value*), 0, 10), **above**(*date*, 0, 10))

rangexirr(**above**(**total** *value*,0,**rowno**(**total**)),

above(**total** *date*,0,**rowno**(**total**)))

rangexnpv(*rate*, *value*, *date* {,*value*, *date*})

Returns the net present value for a schedule of cash flows that is not necessarily periodic. The result has a default number format of money. To calculate the net present value for a series of periodic cash flows, use the **rangenpv** function.

Rate is the interest rate per period.

Value is a cash flow or a series of cash flows that corresponds to a schedule of payments in dates. Each *value* may be a single value or a range of values as returned by an inter-record function with a third optional parameter (see page 388). Text values, null values and missing values are disregarded. All payments are discounted based on a 365-day year. The series of values must contain at least one positive and one negative value.

Date is a payment date or a schedule of payment dates that corresponds to the cash flow payments.

Examples:

rangexnpv(0.1, -2500,'2008-01-01',2750,'2008-09-01') returns 80,25

rangexnpv(0.1, **above**(sum(value), 0, 10), **above**(date, 0, 10))

rangexnpv(0.1, **above**(total value,0,rowno(total)), **above**(total date,0,rowno(total)))

Exponential and Logarithmic Functions

exp(*x*) Exponential function, with the base of the natural logarithm *e* as base. The result is a positive number.

log(*x*) The natural logarithm of *x*. The function is only defined if $x > 0$. The result is a number.

log10(*x*)
The 10-logarithm (base 10) of *x*. The function is only defined if $x > 0$. The result is a number.

sqrt(*x*) Square root of *x*. The function is only defined if $x \geq 0$. The result is a positive number.

sqr(*x*) Square of *x*. The result is a number.

pow(*x,y*)
Returns *x* to the power of *y*. The result is a number.

Trigonometric and Hyperbolic Functions

All angles are measured in radians.

cos(*x*) Cosine of *x*. The result is a number between -1 and 1.

acos(x)

Arcus cosine of x . The function is only defined if $-1 \leq x \leq 1$. The result is a number between 0 and π .

sin(x) Sine of x . The result is a number between -1 and 1.

asin(x)

Arcus sine of x . The function is only defined if $-1 \leq x \leq 1$. The result is a number between $-\pi/2$ and $\pi/2$.

tan(x) Tangent of x . The result is a number.

atan(x)

Arcus tangent of x . The result is a number between $-\pi/2$ and $\pi/2$.

atan2(y,x)

Two-dimensional generalization of the arcus tangent function. Returns the angle between the origin and the point represented by the coordinates x and y . The result is a number between $-\pi$ and $+\pi$.

cosh(x)

Cosine hyperbolicus of x . The result is a positive number.

sinh(x)

Sine hyperbolicus of x . The result is a number.

tanh(x)

Tangent hyperbolicus of x . The result is a number.

Statistical Distribution Functions

The statistical distribution functions described below are all implemented in QlikView using the Cephess library. For references and details on algorithms used, accuracy etc., please see <http://www.netlib.org/cephes/>. The Cephess function library is used by permission.

chidist ($value, degrees_freedom$)

returns the one-tailed probability of the chi-squared distribution.

The X^2 distribution is associated with a X^2 test. *Value* is the value at which you want to evaluate the distribution. *Value* must not be negative. *Degrees_freedom* is a positive integer stating the number of degrees of freedom. Both arguments must be numeric, else null will be returned. This function is related to the **chiinv** function in the following way:

If $prob = \mathbf{chidist}(value, df)$, then $\mathbf{chiinv}(prob, df) = value$.

Example:

$\mathbf{chidist}(8, 15)$ returns 0.9237827

chiinv (*prob, degrees_freedom*)

returns the inverse of the one-tailed probability of the chi-squared distribution. *Prob* is a probability associated with the chi-squared distribution. It must be a number between 0 and 1. *Degrees_freedom* is an integer stating the number of degrees of freedom. Both arguments must be numeric, else null will be returned. This function is related to the **chidist** function in the following way:

If $prob = \mathbf{chidist}(value, df)$, then $\mathbf{chiinv}(prob, df) = value$.

Example:

$\mathbf{chiinv}(0.9237827, 15)$ returns 8.0000001

normdist (*value, mean, standard_dev*)

returns the cumulative normal distribution for the specified mean and standard deviation. *Value* is the value at which you want to evaluate the distribution. *Mean* is a value stating the arithmetic mean for the distribution. *Standard_dev* is a positive value stating the standard deviation of the distribution. All arguments must be numeric, else null will be returned. If $mean = 0$ and $standard_dev = 1$, the function returns the standard normal distribution. This function is related to the **norminv** function in the following way:

If $prob = \mathbf{normdist}(value, m, sd)$, then $\mathbf{norminv}(prob, m, sd) = value$.

Example:

$\mathbf{chidist}(0.5, 0, 1)$ returns 0.6914625

norminv (*prob, mean, standard_dev*)

returns the inverse of the normal cumulative distribution for the specified mean and standard deviation. *Prob* is a probability associated with the normal distribution. It must be a number between 0 and 1. *Mean* is a value stating the arithmetic mean for the distribution. *Standard_dev* is a positive value stating the standard deviation of the distribution. All arguments must be numeric, else null will be returned. This function is related to the **normdist** function in the following way:

If $prob = \mathbf{normdist}(value, m, sd)$, then $\mathbf{norminv}(prob, m, sd) = value$.

Example:

$\mathbf{norminv}(0.6914625, 0, 1)$ returns 0.4999717

tdist (*value*, *degrees_freedom*, *tails*)

returns the probability for the Student t-distribution where a numeric value is a calculated value of t for which the probability is to be computed. *Value* is the value at which you want to evaluate the distribution and must not be negative. *Degrees_freedom* is a positive integer stating the number of degrees of freedom. *Tails* must be either 1 (one-tailed distribution) or 2 (two-tailed distribution). All arguments must be numeric, else null will be returned. This function is related to the **tinvs** function in the following way:

If $prob = \mathbf{tdist}(value, df, 2)$, then $\mathbf{tinvs}(prob, df) = value$.

Example:

$\mathbf{tdist}(1, 30, 2)$ returns 0.3253086

tinvs (*prob*, *degrees_freedom*)

returns the t-value of the Student's t-distribution as a function of the probability and the degrees of freedom. *Prob* is a two-tailed probability associated with the t-distribution. It must be a number between 0 and 1. *Degrees_freedom* is an integer stating the number of degrees of freedom. Both arguments must be numeric, else null will be returned. This function is related to the **tdist** function in the following way:

If $prob = \mathbf{tdist}(value, df, 2)$, then $\mathbf{tinvs}(prob, df) = value$.

Example:

$\mathbf{tinvs}(0.3253086, 30)$ returns 1

fdist (*value*, *degrees_freedom1*, *degrees_freedom2*)

returns the F-distribution.. *Value* is the value at which you want to evaluate the distribution. *Value* must not be negative. *Degrees_freedom1* is a positive integer stating the number of numerator degrees of freedom. *Degrees_freedom2* is a positive integer stating the number of denominator degrees of freedom. All arguments must be numeric, else null will be returned. This function is related to the **finvs** function in the following way:

If $prob = \mathbf{fdist}(value, df1, df2)$, then $\mathbf{finvs}(prob, df1, df2) = value$.

Example:

$\mathbf{fdist}(15, 8, 6)$ returns 0.0019369

finvs (*prob*, *degrees_freedom1*, *degrees_freedom2*)

returns the inverse of the F-distribution. *Prob* is a probability associated with the F-distribution and must be a number between 0 and 1. *Degrees_freedom1* is a positive integer stating the number of numerator degrees of freedom. *Degrees_freedom2* is a positive integer stating the number of denominator degrees of freedom. All argu-

ments must be numeric, else null will be returned. This function is related to the **fdist** function in the following way:

If $prob = \mathbf{fdist}(value, df1, df2)$, then $\mathbf{finv}(prob, df1, df2) = value$.

Example:

finv(0.0019369, 8, 5) returns 15.0000197

Financial Functions

Note When using the financial functions described below, it is vital to be consistent about the units you use for specifying *rate* and *nper*. If you make monthly payments on a five-year loan at 6% annual interest, use 0.005 (6%/12) for *rate* and 60 (5*12) for *nper*. If you make annual payments on the same loan, use 6% for *rate* and 5 for *nper*. In all the functions, cash you pay out is represented by negative numbers. Cash you receive is represented by positive numbers.

BlackAndSchole(*strike, time_left, underlying_price, vol, risk_free_rate, type*)

Returns the theoretical value of an option according to Black and Schole's unmodified formula (European style options).

Strike is the future purchase price of the stock.

Time_left is the number of time periods remaining.

Underlying_price is the current value of the stock.

Vol is the volatility in % per time period.

Risk_free_rate is the risk free rate in % per time period.

Type is 'c', 'call' or any non-zero numeric value for call-options and 'p', 'put' or 0 for put-options..

Example:

The theoretical price of an option to buy in 4 years at a value of 130 a share which is today worth 68.5 assuming a volatility of 40% per year and a risk-free interest rate of 4%?

blackandschole(130, 4, 68.5, 0.4, 0.04, 'call') returns 11.245...

The following approximation is used to calculate the inverse of the normal cumulative distribution in the Black and Schole formula:

For $x > 0$

$$\int_{-\infty}^x e^{-u^2/2} du \approx \sqrt{2\pi} - (b_1 k + b_2 k^2 + b_3 k^3) e^{-x^2/2}$$

$$k = \frac{1}{1 + 0.33267x} \quad b = \begin{pmatrix} 0.4361836 \\ -0.1201676 \\ 0.937298 \end{pmatrix}$$

For $x < 0$

$$f(x) = 1 - f(-x).$$

fv(*rate*, *nper*, *pmt* [, *pv* [, *type*]])

Returns the future value of an investment based on periodic, constant payments and a constant interest rate. The result has a default number format of money.

Rate is the interest rate per period.

Nper is the total number of payment periods in an annuity.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *pv* argument.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now. If *pv* is omitted, it is assumed to be 0 (zero), and you must include the *pmt* argument.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Example:

You pay a new VCR by 36 monthly installments of \$20. The interest rate is 6% per annum. The bill comes at the end of the month. What is the total value of the money you invested when the last bill has been paid?

fv(0.005, 36, -20) returns \$786.72

nper(*rate*, *pmt*, *pv* [, *fv* [, *type*]])

Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.

Rate is the interest rate per period.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *pv* argument.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, it is assumed to be 0.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Example:

You want to sell a VCR by monthly installments of \$20. The interest rate is 6% per annum. The bill comes at the end of the month. How many periods are required if value of the money you have received after the last bill has been paid should equal \$786.72?

nper(0.005,-20,0,786.72) returns 36

pmt(rate, nper, pv [, fv [, type]])

Returns the payment for a loan based on constant payments and a constant interest rate. The result has a default number format of money.

Rate is the interest rate per period.

Nper is the total number of payment periods in an annuity.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, it is assumed to be 0.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

To find the total amount paid over the duration of the loan, multiply the returned *pmt* value by *nper*.

Examples:

The following formula returns the monthly payment on a \$20,000 loan at an annual rate of 10 percent that you must pay off in 8 months:

pmt(0.1/12, 8, 20000) returns -\$2594.66

For the same loan, if payments are due at the beginning of the period, the payment is:

pmt(0.1/12, 8, 20000, 0, 1) returns -\$2573.21

pv(*rate*, *nper*, *pmt* [, *fv* [, *type*]])

Returns the present value of an investment. The result has a default number format of money. The present value is the total amount that a series of future payments is worth now. For example, when you borrow money, the loan amount is the present value to the lender.

Rate is the interest rate per period.

Nper is the total number of payment periods in an annuity.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *fv* argument.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, you must include the *pmt* argument.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Example:

What is the present value of \$100 paid to you at the end of each month during a five year period? The interest rate is 7%.

pv(0.07/12, 12*5, 100, 0, 0) returns -\$5050.20

rate(*nper*, *pmt*, *pv* [, *fv* [, *type*]])

Returns the interest rate per period of an annuity.

Nper is the total number of payment periods in an annuity.

Pmt is the payment made each period. It cannot change over the life of the annuity. If *pmt* is omitted, you must include the *fv* argument.

Pv is the present value, or the lump-sum amount that a series of future payments is worth right now.

Fv is the future value, or a cash balance you want to attain after the last payment is made. If *fv* is omitted, it is assumed to be 0.

Type should be 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

Rate is calculated by iteration and can have zero or more solutions. If the successive results of **rate** do not converge, a null value will be returned.

Example:

What is the interest rate of a five-year \$10,000 annuity loan with monthly payments of \$300?

rate(60,-300,10000) returns 2.18%

This is the monthly rate, which has to be multiplied with 12 in order to get the annual rate, in this case 26.1%.

Mathematical Constants and Parameter-Free Functions

These functions do not have any parameters. The parentheses are however still needed.

e() Base of the natural logarithms, e. The function returns 2.71828...

pi() The function returns 3.14159...

rand() Returns a random number between 0 and 1.

true() Returns a dual value with text value 'true' and numeric value -1, which can be used as logical true in expressions.

false() Returns a dual value with text value 'false' and numeric value 0, which can be used as logical false in expressions.

String Functions

ord(s)

ASCII number of first character of string *s*. The result is an integer.

Example:

ord('A') returns the number 65

chr(n) The ASCII character corresponding to number *n*. The result is a string.

Example:

chr(65) returns the string 'A'

len(s) Length of string *s*. The result is an integer.

Example:

len(Name) where Name = 'Peter' returns 5.

left(*s* , *n*)

Substring of the string *s*. The result is a string consisting of the first *n* characters of *s*.

Examples:

left('abcdef',3) returns 'abc'

left(Date, 4) where Date = 1997-07-14 returns 1997

For a more complicated example, see the **index** function below.

right(*s* , *n*)

Substring of the string *s*. The result is a string consisting of the last *n* characters of *s*.

Examples:

right('abcdef',3) returns 'def'

right(Date,2) where Date = 1997-07-14 returns 14.

mid(*s* , *n1* (, *n2*))

Substring of the string *s*. The result is the string starting at character *n1* with the length of *n2* characters. If *n2* is omitted, the function returns the rightmost part of the string starting at character *n1*. The positions in the string are numbered from 1 and up.

Examples:

mid('abcdef',3) returns 'cdef'

mid('abcdef',3, 2) returns 'cd'

mid(Date,3) where Date = 970714 returns 0714

mid(Date,3,2) where Date = 970714 returns 07

For a more complicated example, see the **index** function below.

index(*s1* , *s2* [, *n*])

Position of a substring. This function gives the starting position of the *n*:th occurrence of substring *s2* in string *s1*. If *n* is omitted, the first occurrence is assumed. If *n* is negative, the search is made starting from the end of string *s1*. The result is an integer. The positions in the string are numbered from 1 and up.

Examples:

index('abcdefg', 'cd') returns 3

index('abcdabcd', 'b', 2) returns 6

index('abcdabcd', 'b', -2) returns 2

left(Date, index(Date, '-') -1)

where Date = 1997-07-14 returns 1997

mid(Date, index(Date, '-', 2) -2, 2)

where Date = 1997-07-14 returns 07

upper(*text expression*)

Forces upper case for all the data in the expression.

Example:

upper('abcD') returns 'ABCD'

lower(*text expression*)

Forces lower case for all the data in the expression.

Example:

lower('abcD') returns 'abcd'

capitalize(*s*)

Returns the string *s* with all words capitalized.

Example:

capitalize('my little pony') returns 'My Little Pony'

capitalize('AA bb cC Dd') returns 'Aa Bb Cc Dd'

repeat(*s, n*)

Forms a string consisting of the string *s* repeated *n* times.

Example:

repeat('*', rating) where rating = 4 returns '****'

ltrim(*s*)

Returns the string *s* trimmed of any leading spaces.

Examples:

ltrim(' abc') returns 'abc'

ltrim('abc ') returns 'abc '

rtrim(*s*)

Returns the string *s* trimmed of any trailing spaces.

Examples:

rtrim(' abc') returns ' abc'

rtrim('abc ') returns 'abc'

trim(*s*)

Returns the string *s* trimmed of any leading and trailing spaces.

Examples:

trim(' abc') returns 'abc'

trim('abc ') returns 'abc'

trim(' abc ') returns 'abc'

KeepChar(*s1*, *s2*)

Returns the string *s1* less all characters not contained in string *s2*.

Example:

keepchar('a1b2c3' , '123') returns '123'

PurgeChar(*s1*, *s2*)

Returns the string *s1* less all characters contained in string *s2*.

Example:

purgechar('a1b2c3' , '123') returns 'abc'.

replace (*string*, *fromstring*, *tostring*)

Returns a *string* after replacing all occurrences of a given substring within the string with another substring. The function is non-recursive and works from left to right.

String is the original string.

fromstring is a string which may occur one or more times within *string*.

tostring is the string which will replace all occurrences of *fromstring* within *string*.

Example:

replace('abcde','cc','xyz')

subfield(*s*, '*delimiter*', *index*)

This chart function returns a given substring from a larger string (*s*) with delimiters ('*delimiter*'). *Index* is an optional integer denoting which of the substrings should be returned.

Examples:

subfield(S, ';' ,2) returns 'cde' if S is 'abc;cde;efg'

subfield(S, ':' ,1) returns NULL if S is an empty string

subfield(S, ':' ,1) returns an empty string if S is ':'

info(*field*)

Returns external info as linked by an **info load** or **info select** statement for a given field. If no external info is available for the field, NULL is returned. The external info is returned in the form of a text string. If the info item is a file, the file name is returned as a string.

TextBetween (*s*, *beforetext*, *aftertext* [, *n*])

Returns the text between the *n*:th occurrence of *beforetext* and the immediately following occurrence of *aftertext* within the string *s*.

Examples:**TextBetween**('<abc>', '<', '>') returns 'abc'**TextBetween**('<abc><de>', '<', '>', 2) returns 'de'**findoneof** (*text*, *characterset* [, *n*])

returns the position of the *n*th occurrence in the string *text* of any of the characters found in the string *characterset*. If *n* is omitted the position of the first occurrence is returned. If there is no match 0 is returned.

Examples:**findoneof**('my example text string', 'et%s') returns 4**findoneof**('my example text string', 'et%s', 3) returns 12**findoneof**('my example text string', 'æ%s&') returns 0**substringcount**(*text*, *substring*)

Returns the number of times the string *substring* appears within the string *text*. The result is an integer. If there is no match, 0 is returned.

Example:**substringcount** ('abcdefgcdxyz', 'cd') returns 2**applycodepage**(*codepage*, *text*)

Applies a different codepage to the field or text stated in the expression. The codepage must be in number format.

Inter-Record Functions

fieldvalue(*fieldname*, *n*)

Returns the field value found in position *n* of the field *fieldname* (by load order). *Fieldname* must be given as a string value, e.g. the field name must be enclosed by single quotes. The first field value is returned for *n*=1. If *n* is larger than the number of field values, NULL is returned.

Example:**fieldvalue**('Helptext', 5)**fieldvaluecount** (*fieldname*)

Returns the number of distinct values in a field. *Fieldname* must be given as a string (e.g. a quoted literal).

Example:

fieldvaluecount('Name')

Column(*ColumnNo*)

Returns the value found in the column *ColumnNo* in a straight table or a pivot table.

Example:

column(1)/column(2) returns the quotient in a new column in the table.

fieldindex(fieldname, value)

Returns the position of the field value in the field fieldname (by load order). If value cannot be found among the field values of the field fieldname, 0 is returned. Fieldname must be given as a string value, e.g. the field name must be enclosed by single quotes.

Note The position of the field's first field value is 1. The use of this function in chart expressions on fields with many field values may be somewhat expensive in terms of memory usage.

Example:

fieldindex('Name', 'John Doe')

above([**total**] *expression* [, *offset* [, *n*]])

Returns the value of *expression* evaluated with the chart's dimension values as they appear on the row above the current row within a column segment in a table or, in the case of image charts, in the chart's straight table equivalent.

On the first row of a column segment a null value will be returned, as there is no row above.

If the chart is one-dimensional or if the expression is preceded by the qualifier **total**, the current column segment is always equal to the entire column.

If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The inter field sort order for pivot tables is defined simply by the order of the dimensions from the left to the right, while for other

chart types it can be manipulated on the **Sort** page of the **Chart Properties** dialog.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved up to rows further above the current row. E.g. if *offset* is 2, the function returns the value of *expression* evaluated with the chart's dimensions as they appear two rows above the current row.

A negative *offset* makes the **above** function equivalent to the **below** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* rows of the table going upwards from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

sum(Sales)/above(sum(Sales))

above(sum(Sales), 2)

above(total sum(Sales))

rangeavg (above(sum(x),1,3,)) returns an average of the three results of the sum(x) function evaluated on the three rows immediately above the current row.

More detailed examples on the use of chart inter-record functions are found on page 453.

below([total] expression [, offset [,n]])

Returns the value of *expression* evaluated with the chart's dimension values as they appear on the row below the current row within a column segment in a table or, in the case of image charts, in the chart's straight table equivalent.

On the last row of a column segment a null value will be returned, as there is no row below.

If the chart is one-dimensional or if the is preceded by the qualifier **total**, the current column segment is always equal to the entire column.

If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The inter field sort order for pivot tables is defined simply by the order of the dimensions from the left to the right, while for other chart types it can be manipulated on the **Sort** page of the **Chart Properties** dialog.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved down to rows further below the current row. E.g. if offset is 2, the function returns the value of *expression* evaluated with the chart's dimensions as they appear two rows below the current row.

A negative *offset* makes the **below** function equivalent to the **above** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* rows of the table going downwards from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

below(sum(Sales))

below(sum(Sales), 2)

below(total sum(Sales))

rangeavg (below(sum(x),1,3)) returns an average of the three results of the sum(x) function evaluated on the three rows immediately below the current row.

More detailed examples on the use of chart inter-record functions are found on page 453.

top([total] expression [, offset [,n]])

Returns the value of *expression* evaluated with the chart's dimension values as they appear on the first row of the column segment in a table or, in the case of image charts, in the chart's straight table equivalent.

If the chart is one-dimensional or if the *expression* is preceded by the qualifier **total**, the current column segment is always equal to the entire column.

If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The inter field sort order for pivot tables is defined simply by the order of the dimensions from the left to the right, while for other chart types it can be manipulated on the **Sort** page of the **Chart Properties** dialog.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved down to rows below the top row. E.g. if *offset* is 2, the function returns the value of *expression* evaluated with the chart's dimensions as they appear on the second row as defined above.

A negative *offset* makes the **top** function equivalent to the **bottom** function with the corresponding positive *offset*. An offset of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* rows of the table going downwards from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

The **top** function will only be evaluated in total rows if the total in question has a clear relation to one specific column segment. In all other cases NULL will be returned.

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

sum(Sales)/top(sum(Sales))

top(sum(Sales), 2)

top(total sum(Sales))

rangeavg (top(sum(x),1,5)) returns an average of the results of the sum(x) function evaluated on the five top rows of the table.

More detailed examples on the use of chart inter-record functions are found on page 453.

bottom([total] expression [, offset[,n]])

Returns the value of *expression* evaluated with the chart's dimension values as they appear on the last row of the column segment in a table or, in the case of image charts, in the chart's straight table equivalent.

If the chart is one-dimensional or if the *expression* is preceded by the qualifier **total**, the current column segment is always equal to the entire column.

If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The inter field sort order for pivot tables is defined simply by the order of the dimensions from the left to the right, while for other chart types it can be manipulated on the **Sort** page of the **Chart Properties** dialog.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved up to rows above the bottom row. E.g. if *offset* is 2,

the function returns the value of *expression* evaluated with the chart's dimensions as they appear on the second row from the bottom as defined above.

A negative *offset* makes the **bottom** function equivalent to the **top** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one value but a range of *n* values, one for each of *n* rows of the table going upwards from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

The **bottom** function will only be evaluated in total rows if the total in question has a clear relation to one specific column segment. In all other cases NULL will be returned.

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

bottom(sum(Sales))

bottom(sum(Sales), 2)

bottom(total sum(Sales))

rangeavg (bottom(sum(x),1,5)) returns an average of the results of the sum(x) function evaluated on the five bottom rows of the table.

More detailed examples on the use of chart inter-record functions are found on page 453.

before([total] expression [, offset [,n]])

Returns the value of *expression* evaluated with a pivot table's dimension values as they appear in the column before the current column within a row segment in the pivot table. This function returns NULL in all chart types except pivot tables.

In the first column of a row segment a null value will be returned, as there is no column before it.

If the table is one-dimensional or if the *expression* is preceded by the qualifier **total**, the current row segment is always equal to the entire row.

If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The inter field sort order for horizontal dimensions in pivot tables is defined simply by the order of the dimensions from top to bottom.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved to columns further to the left from the current column. E.g. if *offset* is 2, the function returns the value of *expression* evaluated with the chart's dimensions as they appear two columns to the left from the current column.

A negative *offset* makes the **before** function equivalent to the **after** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* columns of the table going to the left from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

before(sum(Sales))

before(sum(Sales), 2)

before(total sum(Sales))

rangeavg(before(sum(x),1,3)) returns an average of the three results of the sum(x) function evaluated in the three columns immediately to the left of the current column.

More detailed examples on the use of chart inter-record functions are found on page 453.

after([total] *expression* [, *offset* [, *n*]])

Returns the value of *expression* evaluated with a pivot table's dimension values as they appear in the column after the current column within a row segment in the pivot table. This function returns NULL in all chart types except pivot tables.

In the last column of a row segment a null value will be returned, as there is no column after it.

If the table is one-dimensional or if the expression is preceded by the qualifier **total**, the current row segment is always equal to the entire row.

If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The inter field sort order for horizontal dimensions in pivot tables is defined simply by the order of the dimensions from top to bottom.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved to columns further to the right from the current column. E.g. if *offset* is 2, the function returns the value of *expression* evaluated with the chart's dimensions as they appear two columns to the right from the current column.

A negative *offset* makes the **after** function equivalent to the **before** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* columns of the table going to the right from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

after(**sum**(Sales))

after(**sum**(Sales), 2)

after(**total sum**(Sales))

rangeavg (after(sum(x),1,3)) returns an average of the three results of the sum(x) function evaluated in the three columns immediately to the right of the current column.

More detailed examples on the use of chart inter-record functions are found on page 453.

first([total] expression [, offset [,n]])

Returns the value of *expression* evaluated with a pivot table's dimension values as they appear in the first column of the current row segment in the pivot table. This function returns NULL in all chart types except pivot tables.

If the table is one-dimensional or if the expression is preceded by the qualifier **total**, the current row segment is always equal to the entire row.

If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The inter field sort order for horizontal dimensions in pivot tables is defined simply by the order of the dimensions from top to bottom. By specifying an *offset* greater than 1, the evaluation of *expression* may be moved to columns on the right-hand side of the first column. E.g. if *offset* is 2, the function returns the value of expression evaluated with the chart's dimensions as they appear in the second column as defined above.

A negative *offset* makes the **first** function equivalent to the **last** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* columns of the table going to the right from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

The **first** function will only be evaluated in total rows if the total in question has a clear relation to one specific row segment. In all other cases NULL will be returned.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

first(sum(Sales))

first(sum(Sales), 2)

first(total sum(Sales))

rangeavg (first(sum(x) ,1,5)) returns an average of the results of the sum(x) function evaluated on the five leftmost columns of the table.

More detailed examples on the use of chart inter-record functions are found on page 453.

last([total] expression [, offset [,n]])

Returns the value of *expression* evaluated with a pivot table's dimension values as they appear in the last column of the current row segment in the pivot table. This function returns NULL in all chart types except pivot tables.

If the table is one-dimensional or if the *expression* is preceded by the qualifier **total**, the current row segment is always equal to the entire row.

If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The inter field sort order for horizontal dimensions in pivot tables is defined simply by the order of the dimensions from top to bottom.

By specifying an *offset* greater than 1, the evaluation of *expression* may be moved to columns on the left-hand side of the first column. E.g. if *offset* is 2, the function returns the value of *expression* evaluated with the chart's dimensions as they appear in the second last column as defined above.

A negative *offset* makes the **last** function equivalent to the **last** function with the corresponding positive *offset*. An *offset* of 0 will evaluate the expression on the row of the expression itself. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1 the function will return not one value but a range of *n* values, one for each of *n* col-

umns of the table going to the left from the cell originally pointed at by the function. The function can in this form be used as an argument to any of the special range functions (see page 367).

The **last** function will only be evaluated in total rows if the total in question has a clear relation to one specific row segment. In all other cases NULL will be returned.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

last(sum(Sales))

last(sum(Sales), 2)

last(total sum(Sales))

rangeavg (last(sum(x),1,5)) returns an average of the results of the sum(x) function evaluated on the five rightmost columns of the table.

More detailed examples on the use of chart inter-record functions are found on page 453.

RowNo([total])

Returns the number of the current row within the current column segment in a table or, in the case of image charts, within the chart's straight table equivalent. The first row has number 1.

If the chart is one-dimensional or if the qualifier **total** appears as argument, the current column segment is always equal to the entire column.

If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The inter field sort order for pivot tables is defined simply by the order of the dimensions from the left to the right, while for other chart types it can be manipulated on the **Sort** page of the **Chart Properties** dialog.

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

if(rowno()=1, 0, sum(Sales)/above(sum(Sales)))

More detailed examples on the use of chart inter-record functions are found on page 453.

ColumnNo([total])

Returns the number of the current row within the current row segment in a pivot table. The first column has number 1.

If the table is one-dimensional or if the qualifier **total** appears as argument, the current row segment is always equal to the entire row.

If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The inter field sort order for horizontal dimensions in pivot tables is defined simply by the order of the dimensions from top to bottom.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

if(columnno()=1, 0, sum(Sales)/before(sum(Sales)))

More detailed examples on the use of chart inter-record functions are found on page 453.

NoOfRows([total])

Returns the number of rows within the current column segment in a table or, in the case of image charts, within the chart's straight table equivalent.

If the chart is one-dimensional or if the qualifier **total** appears as argument, the current column segment is always equal to the entire column.

If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The inter field sort order for pivot tables is defined simply by the order of the dimensions from the left to the right, while for other chart types it can be manipulated on the **Sort** page of the **Chart Properties** dialog.

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Note Suppression of zero values is automatically disabled when this function is used.

Examples:

```
if( rowno(=noofrows(), 0, after( sum( Sales )))
```

More detailed examples on the use of chart inter-record functions are found on page 453.

NoOfColumns([total])

Returns the number of columns in the current row segment of a pivot table.

If the table is one-dimensional or if the qualifier **total** appears as argument, the current row segment is always equal to the entire row.

If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The inter field sort order for horizontal dimensions in pivot tables is defined simply by the order of the dimensions from top to bottom.

Note Sorting on Y-values in charts and sorting by expression columns in straight tables is not allowed when using this function in any of the chart's expressions.

Examples:

if(columnno()=noofcolumns(), 0, after(sum(Sales)))

More detailed examples on the use of chart inter-record functions are found on page 453.

dimensionality ()

returns the number of dimension columns that have non-aggregation content, i.e. do not contain partial sums or collapsed aggregates.

A typical use is in attribute expressions, when you want to apply different cell formatting depending on aggregation level of data.

This function is only available in charts. For all chart types except pivot tables it will return the number of dimensions in all rows except the total, which will be 0.

Example:

For a pivot table with three dimension columns to the left the following would be returned:

3 for all ordinary data cells

2 for 1st level partial sums and entries not expanded in the 3rd column

1st for 2nd level partial and entries only expanded in the 1st column.

0 for grand total cell.

Country	Team	Salesman	dimensionality()
Denmark	Copenhagen	Preben	3
		Total	2
	Total		1
Germany			1
Sweden	Gothenburg	John	3
		Total	2
	Lund		2
	Total		1
Total			0

Country	Team	Salesman	dimensionality()
			0
Denmark	Copenhagen	Preben	3
Germany	Berlin	Helga	3
Sweden	Gothenburg	John	3
Sweden	Lund	Ann	3
Sweden	Lund	Peter	3

secondarydimensionality ()

returns the number of dimension pivot table rows that have non-aggregation content, i.e. do not contain partial sums or collapsed aggregates. This function is the equivalent of the dimensionality () function for horizontal pivot table dimensions.

The secondarydimensionality() function always returns 0 when used outside of pivot tables.

Example

Country	Team	Salesman	dimensionality()
			0
		Total	1
		Total	2
	Copenhagen		3
			1

Document Functions

ReportComment(*report_number*)

Returns the comment of the report with the specified number within the active document.

ReportName(*report_number*)

Returns the name of the report with the specified number within the active document.

ReportID(*report_number*)

Returns the id of the report with the specified number within the active document.

ReportNumber(*report_id_or_name*)

Returns the number of the report with the specified id or name within the active document.

NoOfReports()

Returns the number of reports in the active document.

Ranking Functions

rank([*total*] *expression* [, *mode* [, *format*]])

Evaluates *expression*, compares the result with the result on the other rows within the current column segment and returns the ranking of the current row within the segment. For image charts the current column segment is defined as it appears in the chart's straight table equivalent.

If the chart is one-dimensional or if the expression is preceded by the qualifier **total**, the current column segment is always equal to the entire column. If the table or table equivalent has multiple vertical dimensions, the current column segment will include only rows with the same values as the current row in all dimension columns except for the column showing the last dimension in the inter field sort order.

The ranking is returned as a dual value, which in the case where each row has a unique ranking will be an integer between 1 and the number of rows in the current column segment.

In a case where several rows share the same ranking, the text and number representations can be controlled as follows:

The second parameter *mode* specifies the number representation of the function result

mode

0 (default)

If all ranks within the sharing group fall on the low side of the middle value of the entire ranking, all rows get the lowest rank within the sharing group.

If all ranks within the sharing group fall on the high side of the middle value of the entire ranking, all rows in the group get the highest rank within the sharing group.

If ranks within the sharing group span over the middle value of the entire ranking, all rows in the group get a value corresponding to the average of the top and bottom ranking in the entire column segment.

- 1 lowest rank on all rows in the group
- 2 average rank on all rows in the group
- 3 highest rank on all rows in the group
- 4 lowest rank on first row, then increment by one for each row in the group

The third parameter format specifies the text representation of the function result

format

0 (default)

low value&'-'&high value on all rows in the group (e.g. '3-4')

1

low value on all rows in the group

2

low value on first row, blank on the following rows in the group

The order of rows for mode 4 and format 2 is determined by the sort order of the chart dimensions.

Note Suppression of zero values is automatically disabled when this function is used. Null values are disregarded.

Examples:

rank(sum(Sales))

rank(sum(Sales), 2)

rank(sum(Sales), 0, 1)

More detailed examples of the use of ranking functions are found on page 450.

hrank([**total**] *expression* [, *mode* [, *format*]])

Evaluates *expression*, compares the result with the result in the other columns within the current row segment of a pivot table and returns the ranking of the current column within the segment.

This function works only in pivot tables and returns NULL in all other types of charts.

If the pivot table is one-dimensional or if the expression is preceded by the qualifier **total**, the current row segment is always equal to the entire row. If the pivot table has multiple horizontal dimensions, the current row segment will include only columns with the same values as the current column in all dimension rows except for the row showing the last horizontal dimension of the inter field sort order.

The ranking is returned as a dual value, which in the case where each column has a unique ranking will be an integer between 1 and the number of columns in the current row segment.

In a case where several columns share the same ranking, the text and number representations can be controlled as follows:

The second parameter *mode* specifies the number representation of the function result

mode

0 (default)

If all ranks within the sharing group fall on the low side of the middle value of the entire ranking, all rows in the group get the lowest rank within the sharing group.

If all ranks within the sharing group fall on the high side of the middle value of the entire ranking, all rows in the group get the highest rank within the sharing group.

If ranks within the sharing group span over the middle value of the entire ranking, all rows in the group get a value corresponding to the average of the top and bottom ranking in the entire row segment.

1

lowest rank on all rows in the group

2

average rank on all rows in the group

3

highest rank on all rows in the group

4

lowest rank on first row, then increment by one for each row in the group

The third parameter format specifies the text representation of the function result

format

0 (default)

low value&'-'&high value on all rows in the group (e.g. '3-4')

1

low value on all rows in the group

2

low value on first row, blank on the following rows in the group

The order of rows for mode 4 and format 2 is determined by the sort order of the chart dimensions.

Note Suppression of zero values is automatically disabled when this function is used. Null values are disregarded.

Examples:

hrank(sum(Sales))

hrank(sum(Sales), 2)

hrank(sum(Sales), 0, 1)

More detailed examples of the use of ranking functions are found on page 450.

vrank([total] expression [, mode [, format]])

This function is identical with the **rank** function.

Conditional Functions

if(condition , then [, else[])

The three parameters *condition*, *then* and *else* are all expressions. The first one, *condition*, is interpreted logically. The two other ones, *then* and *else*, can be of any type. They should preferably be of the same type. If *condition* is true, the function returns the value of the expression *then*. If *condition* is false, the function returns the value of the expression *else*. If the *else* expression is omitted and *condition* is false, null is returned.

Example:

```
if( Amount >= 0, 'OK', 'Alarm' )
```

alt(*case1* [, *case2* , *case3* , ...] , *otherwise*)

The **alt** function returns the first of the parameters that has a valid number representation. If no such match is found, the last parameter will be returned. Any number of parameters can be used.

Example:

```
alt(date#( dat,'YYYY/MM/DD' ) ,
```

```
date#( dat,'MM/DD/YYYY' ) ,
```

```
date#( dat,'MM/DD/YY' ) ,
```

```
'No valid date' )
```

will test if the dimension *dat* contains a date according to any of the three specified date formats. If so, it will return the original string and a valid number representation of a date. If no match is found, the text 'No valid date' will be returned (without any valid number representation).

pick(*n* , *expr1* [, *expr2* , ... *exprN*])

Returns the *n*:th expression in the list. *n* is an integer between 1 and *N*.

Examples:

```
pick( 2, 'A', 'B', 4 ) returns 'B'
```

```
pick( 3, 'A', 'B', 4 ) returns 4
```

match(*s* , *expr1* [, *expr2* , ... *exprN*])

Compares the string *s* to a list of strings or string expressions. The result of the comparison is an integer indicating which of the comparison strings/expressions matched. If no match is found, 0 is returned. The match function performs a case sensitive comparison.

Examples:

```
match( M , 'Jan', 'Feb', 'Mar' )
```

where M is 'Feb' returns 2

```
match( M , 'Jan', 'Feb', 'Mar' )
```

where M is 'Apr' or 'jan' returns 0

mixmatch(*s* , *expr1* [, *expr2* , ... *exprN*])

Compares the string *s* to a list of strings or string expressions. The result of the comparison is an integer indicating which of the comparison strings/expressions matched. If no match is found, 0 is

returned. The **mixmatch** function performs a case insensitive comparison.

Example:

mixmatch(M , 'Jan' , 'Feb' , 'Mar')
where M is 'jan' returns 1

wildmatch(*s* , *expr1* [, *expr2* , ...*exprN*])

Compares the string *s* to a list of strings or string expressions. The result of the comparison is an integer indicating which of the comparison strings/expressions that matched. If no match is found, 0 is returned. The wildmatch function performs a case insensitive comparison. The wildcard characters * and ? are allowed in the comparison string.

Examples:

wildmatch(M , 'ja*' , 'fe?' , 'mar')
where M is 'January' returns 1

wildmatch(M , 'ja*' , 'fe?' , 'mar')
where M is 'fex' returns 2

Wildmatch5(*s*,*expr1* [,*expr2*,...*exprN*])

Works as the above wildmatch function, but is used only for compatibility reasons with QlikView version 5.

class(*expression* , *interval* [, *label* [, *offset*]])

Creates a classification of *expressions*. The bin width is determined by the number set as *interval*. The result is shown as $a \leq x < b$, where *a* and *b* are the upper and lower limits of the bin. The *x* can be replaced by an arbitrary string stated in *label*. 0 is normally the default starting point of the classification. This can be changed by adding an *offset*.

Examples:

class(var,10) with var = 23 returns '20<=x<30'

class(var,5,'value') with var = 23 returns '20<= value <25'

class(var,10,'x',5) with var = 23 returns '15<=x<25'

Logical Functions

IsNum(*expr*)

Returns -1 (true) if the expression can be interpreted as a number, otherwise 0 (false).

IsText(*expr*)

Returns -1 (true) if the expression has a text representation, otherwise 0 (false).

Null Functions

Null() Returns a real null value.

IsNull(*expr*)

Returns -1 (true) if *expr* returns null, otherwise 0 (false).

System Functions**OSuser()**

Returns a string containing the name of the current user as returned by the operating system.

QVuser()

Returns a string containing the name of the current QlikView user as defined in a **section access**.

qlikviewversion()

Returns the full QlikView version and build number (e.g. 7.52.3797.0409.3) as a string. This function is only available in QlikView builds after 7.52.3795.

ComputerName()

Returns a string containing the name of the computer as returned by the operating system.

ReloadTime()

Returns a timestamp for the last finished execution of the script.

GetActiveSheetID()

Returns a string containing the Id of the active sheet.

GetRegistryString(*path*, *key*)

Returns the value of a named registry *key* with a given registry *path*.

Example:

```
getregistrystring('HKEY_LOCAL_MACHINE\SOFTWARE\QlikTech\QlikViewServer\Settings 7','EnableSessionLog')
```

SqlValue(*datasource*, *selectstatement* [, *type*])

Returns the value in the first column of the first row returned by **selectstatement**. *datasource* must be a predefined ODBC or OLEDB data source. The third parameter *type* must be either 'ODBC' or 'OLEDB'. If omitted **type='ODBC'** is assumed.

Examples:

```
sqlvalue ('NorthWind', 'select CompanyName from  
Customers where CustomerID='  
&chr(39)&Variable1&chr(39))
```

```
sqlvalue ('NorthWind', 'select CompanyName from  
Customers where CustomerID= ' & chr(39) &  
only(MyQvField) &chr(39) , 'OLEDB' )
```

Note The use of this function can cause delays due to response times in the external database.

GetCurrentField(*groupname*)

returns the name of the field currently active in the group named *groupname*.

Example:

```
getcurrentfield( MyGroup )
```

GetCurrentSelections([*recordsep* [, *tagsep* [, *valuesep* [, *maxvalues*=6]]])

returns a selection stamp of the current selections in the document. *Recordsep* is the separator between lines (fields) in the selection stamp and defaults to chr(13) (carriage return). *Tagsep* is the delimiter between field name and values and defaults to ':'. *Valuesep* is the delimiter between selected values and defaults to ','. *Maxvalues* is the highest number of individual values that will be displayed for any one field before 'x of y' notation is used instead.

Examples:

```
getcurrentselections()
```

```
getcurrentselections (chr(13), '=', ';')
```

```
getcurrentselections (chr(13), '=', ';', 10)
```

GetFieldSelections(*fieldname* [, *valuesep* [, *maxvalues*=6]])

returns a string containing the current selections in a given field.

Valuesep is the delimiter between selected values and defaults to ','.

Maxvalues is the highest number of individual values that will be displayed before 'x of y' notation is used instead.

Examples:

getfieldselections(Country)

getfieldselections (Country, ',')

getfieldselections (Country, ', ' , 10)

GetSelectedCount(*fieldname* [, *includeexcluded=false*])

returns the number of selected values in the field named *fieldname*.

If *includeexcluded* is stated as true, the count will include selected values which are excluded by selections in another field

Examples:

getselectedcount(Customers)

getselectedcount(Customers, true)

GetPossibleCount(*fieldname*)

returns the number of possible values in the field named *fieldname*.

Example:

getpossiblecount(Customers)

GetExcludedCount(*fieldname*)

returns the number of excluded values in the field named *fieldname*.

Example:

getexcludedcount(Customers)

GetAlternativeCount(*fieldname*)

returns the number of alternative values in the field named *fieldname*.

Example:

getalternativecount(Customers)

GetNotSelectedCount(*fieldname* [, *includeexcluded=false*])

returns the number of not-selected values in the field named *fieldname*. The field must be in and-mode for this function to be relevant. If *includeexcluded* is stated as true, the count will include

selected values which are excluded by selections in another field

Examples:

getnotselectedcount(Country)

getnotselectedcount(Country, true)

Date and Time Functions

In the following examples, the default date format YYYY-MM-DD (ISO standard) is assumed. See further in “Number Format Dialogs” on page 301 in Book II for a description of date and time formats in QlikView.

day(date)

Day. The result is a number.

Example:

day(Date) where Date = 1971-10-30 returns 30

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. To solve this problem, you can either change the settings or use the **date#** interpretation function (see under “Number Interpretation Functions” on page 434). For further information about date and time functions see page 296 in Book II.

week(date)

Week number. The result is a number. A week begins on Monday. Week #1 is the first week with four or more days in the new year.

Example:

week(Date) where Date = 1971-10-30 returns 43

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

month(date)

Month. The result is a text string, but can be formatted as a number.

Example:

month(Date) where Date = 1971-10-30 returns Oct

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

year(date)

Year. The result is a number.

Example:

year(Date) where Date = 1971-10-30 returns 1971

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

weekyear(date)

The year to which the week number belongs. The week number ranges between 1 and approximately 52. Some years week #1 starts in December, e.g. December 1997. Other years start with week #53 of previous year, e.g. January 1999. For those few days when the week number belongs to another year, the functions **year** and **week-year** will return different values.

Examples:

weekyear(Date) where Date = 1996-12-30 returns 1997

weekyear(Date) where Date = 1997-01-02 returns 1997

weekyear(Date) where Date = 1997-12-30 returns 1997

weekyear(Date) where Date = 1999-01-02 returns 1998

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

weekday(date)

Week day. The result is a number between 0-6 (0=Monday).

Example:

weekday(Date) where Date = 1971-10-30 returns 5

If the date format used does not correspond to the one set in your operating system, QlikView will not be able to make a correct interpretation. See above under **day**.

MakeDate(YYYY [, MM [, DD]])

Returns a date calculated from the year *YYYY*, the month number *MM* and the day number *DD*. If no month number is stated, 1 (January) is assumed. If no day number is stated, 1 (the 1st) is assumed.

Examples:

(date format YYYY-MM-DD assumed)

makedate(1999) returns 1999-01-01

makedate(99) returns 0099-01-01

makedate(1992, 12) returns 1992-12-01

makedate(1999, 2, 14) returns 1999-02-14

MakeWeekDate(YYYY, WW [, D])

Returns a date calculated from the year *YYYY*, the week number *WW* and the day-of-week *D*, where 0 denotes Monday and 6 denotes Sunday. Week number 1 is the first week containing four days or

more of the year. If no day-of-week is stated, 0 (Monday) is assumed.

Examples:

(date format YYYY-MM-DD assumed)

makeweekdate(1999,6,6) returns 1999-02-14

makeweekdate(99,6) returns 0099-02-02

MakeTime(*hh* [, *mm* [, *ss* [,*.fff*]]])

Returns a time calculated from the hour *hh*, the minute *mm* and the second *ss* with fraction *fff* down to a millisecond. If no minute is stated, 00 is assumed. If no second is stated, 00 is assumed. If no fraction of seconds is stated, .000 is assumed.

Examples:

(time format hh:mm:ss assumed)

maketime(22) returns 22:00:00

maketime(22, 17) returns 22:17:00

maketime(22, 17, 52) returns 22:17:52

hour(*expr*)

Returns an integer representing the hour when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

Examples:

(time format hh:mm:ss assumed)

hour(time) returns 9 when time='09:14:36'

hour(0.5555) returns 13 (0.5555 = 13:19:55)

minute(*expr*)

Returns an integer representing the minute when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

Examples:

(time format hh:mm:ss assumed)

minute(time) returns 14 when time='09:14:36'

minute(0.5555) returns 19 (0.5555 = 13:19:55)

second(*expr*)

Returns an integer representing the second when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

Examples:

(time format hh:mm:ss assumed)

second(time) returns 36 when time='09:14:36'

second(0.5555) returns 55 (0.5555 = 13:19:55)

now([timer_mode])

Returns a timestamp from the system clock. The *timer_mode* may have the following values:

- 0 Time at script run
- 1 Time at function call
- 2 Time when the document was opened

Default *timer_mode* is 1. The *timer_mode=1* should be used with caution, since it polls the operating system every second and hence could slow down the system.

today([timer_mode])

Returns the date from the system clock. The *timer_mode* may have the following values:

- 0 Date at script run
- 1 Date at function call
- 2 Date when the document was opened

Default *timer_mode* is 2. The *timer_mode=1* should be used with caution, since it polls the operating system every second and hence could slow down the system.

LocalTime([timezone [, ignoreDST]])

Returns a timestamp of the current time from the system clock for a specified time zone. The *timezone* is specified as a string containing any of the geographical places listed under Time Zone in the Windows Control Panel for Date and Time or as a string in the form 'GMT+hh:mm'. If no *timezone* is specified the local time will be returned. If *ignoreDST* is true daylight savings time will be ignored.

Examples:

localtime ('Paris')

localtime ('GMT+01:00')

localtime ('Paris', true)

localtime ()

ConvertToLocalTime(timestamp [, place [, ignore_dst=false]])

Converts a UTC or GMT timestamp to local time as a dual value. The place can be any of a number of cities, places and time zones

around the world, e.g. Abu Dhabi, Adelaide, Alaska, Almaty, Amsterdam, Arizona, Astana, Athens, Atlantic Time (Canada), Auckland, Azores, Baghdad, Baku, Bangkok, Beijing, Belgrade, Berlin, Bern, Bogota, Brasilia, Bratislava, Brisbane, Brussels, Bucharest, Budapest, Buenos Aires, Cairo, Canberra, Cape Verde Is., Caracas, Casablanca, Central America, Central Time (US & Canada), Chennai, Chihuahua, Chongqing, Copenhagen, Darwin, Dhaka, Eastern Time (US & Canada), Edinburgh, Ekaterinburg, Fiji, Georgetown, Greenland, Greenwich Mean Time : Dublin, Guadalajara, Guam, Hanoi, Harare, Hawaii, Helsinki, Hobart, Hong Kong, Indiana (East), International Date Line West, Irkutsk, Islamabad, Istanbul, Jakarta, Jerusalem, Kabul, Kamchatka, Karachi, Kathmandu, Kolkata, Krasnoyarsk, Kuala Lumpur, Kuwait, Kyiv, La Paz, Lima, Lisbon, Ljubljana, London, Madrid, Magadan, Marshall Is., Mazatlan, Melbourne, Mexico City, Mid-Atlantic, Midway Island, Minsk, Monrovia, Monterrey, Moscow, Mountain Time (US & Canada), Mumbai, Muscat, Nairobi, New Caledonia, New Delhi, Newfoundland, Novosibirsk, Nuku'alofa, Nuku'alofa, Osaka, Pacific Time (US & Canada), Paris, Perth, Port Moresby, Prague, Pretoria, Quito, Rangoon, Riga, Riyadh, Rome, Samoa, Santiago, Sapporo, Sarajevo, Saskatchewan, Seoul, Singapore, Skopje, Sofia, Solomon Is., Sri Jayawardenepura, St. Petersburg, Stockholm, Sydney, Taipei, Tallinn, Tashkent, Tbilisi, Tehran, Tijuana, Tokyo, Ulaan Bataar, Urumqi, Warsaw, Wellington, West Central Africa, Vienna, Vilnius, Vladivostok, Volgograd, Yakutsk, Yerevan or Zagreb. Also GMT, GMT-01:00, GMT+04:00 etc. are valid places.

The resulting time is adjusted for daylight savings time, unless the third parameter is set to 1 or true().

Examples:

ConvertToLocalTime('2007-11-10 23:59:00', 'Paris') returns '2007-11-11 00:59:00' and the corresponding internal timestamp representation.

ConvertToLocalTime(UTC(), 'GMT-05:00') returns the time for the North American east coast, e.g. New York.

yeartodate(*date* [, *yearoffset* [, *firstmonth* [, *todaydate*]]])

Returns true if *date* falls within the year to date, else false. If none of the optional parameters are used, the year to date means any date within one calendar year from January 1 up to and including the date of the last script execution.

By specifying a *yearoffset* (0 if omitted), the function can be transposed to return true for the same period in another year. Negative *yearoffset* indicates previous years while a positive value indicates coming years. Last year to date is achieved by specifying *yearoffset* = -1.

By specifying a *firstmonth* between 1 and 12 (1 if omitted), the beginning of the year may be moved forward to the first day of any month. If you e.g. want to work with a fiscal year starting May 1, you may specify *firstmonth* = 5.

By specifying a *todaydate* (time of last script execution if omitted), you may move the day used as the upper boundary of the period.

Examples:

```
assume last reload time = 1999-11-18
yeartodate( '1998-11-18' ) returns false
yeartodate( '1999-02-01' ) returns true
yeartodate( '1999-11-18' ) returns true
yeartodate( '1999-11-19' ) returns false
yeartodate( '1998-11-18', -1 ) returns true
yeartodate( '1999-11-18', -1 ) returns false
yeartodate( '1999-04-30', 0, 5 ) returns false
yeartodate( '1999-05-01', 0, 5 ) returns true
```

DayNumberOfYear(*date*[,*firstmonth*])

Returns the day number of the year according to a timestamp with the first millisecond of the first day of the year containing *date*. The function always uses years based on 366 days.

By specifying a *firstmonth* between 1 and 12 (1 if omitted), the beginning of the year may be moved forward to the first day of any month. If you e.g. want to work with a fiscal year starting March 1, you may specify *firstmonth* = 3.

Examples:

DayNumberOfYear(*date*) returns the day number counted from the first of the year.

DayNumberOfYear(*date*,3) returns the number of the day as counted from the first of March.

DayNumberOfQuarter(*date*[,*firstmonth*])

Returns the day number of the quarter according to a timestamp with the first millisecond of the first day of the quarter containing *date*. The function always uses years based on 366 days.

By specifying a *firstmonth* between 1 and 12 (1 if omitted), the beginning of the year may be moved forward to the first day of any month. If you e.g. want to work with a fiscal year starting March 1, you may specify *firstmonth* = 3.

Examples:

DayNumberOfQuarter(Date) returns the day number of the quarter counted from the first day of the first quarter.

DayNumberOfQuarter(Date,3) returns the day number of the quarter counted from the first of March

AddMonths(startdate, n [, mode])

Returns the date occurring *n* months after *startdate* or, if *n* is negative, the date occurring *n* months before *startdate*.

By specifying a *mode* (0 if omitted) the date is set to either the unmodified day of the specified month (*mode*=0) or the calculated day as derived from the end of the month (*mode*=1).

Examples:

addmonths ('2003-01-29',3) returns '2003-04-29'

addmonths ('2003-01-29',3,0) returns '2003-04-29'

addmonths ('2003-01-29',3,1) returns '2003-04-28'

addmonths ('2003-01-29',1,0) returns '2003-02-28'

addmonths ('2003-01-29',1,1) returns '2003-02-26'

addmonths ('2003-02-28',1,0) returns '2003-03-28'

addmonths ('2003-02-28',1,1) returns '2003-03-31'

YearStart(date [, shift = 0 [, first_month_of_year = 1]])

Returns a value corresponding to a timestamp with the first millisecond of the first date of the year containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the year which contains *date*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Example:

yearstart ('2001-10-19') returns '2001-01-01' with an underlying numeric value corresponding to '2001-01-01 00:00:00.000'

yearstart ('2001-10-19', -1) returns '2000-01-01' with an underlying numeric value corresponding to '2000-01-01 00:00:00.000'

yearstart ('2001-10-19', 0, 4) returns '2001-04-01' with an underlying numeric value corresponding to '2001-04-01 00:00:00.000'

YearEnd(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the last millisecond of the last date of the year containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the year which contains *date*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Example:

yearend ('2001-10-19') returns '2001-12-31' with an underlying numeric value corresponding to '2001-12-31 23:59:59.999'

yearend ('2001-10-19', -1) returns '2000-12-31' with an underlying numeric value corresponding to '2000-12-31 23:59:59.999'

yearend ('2001-10-19', 0, 4) returns '2002-03-31' with an underlying numeric value corresponding to '2002-03-31 23:59:59.999'

YearName(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a four-digit year as display value with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the year containing *date*. *Shift* is an integer, where the value 0 indicates the year which contains *date*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*. The display value will then be a string showing two years.

Examples:

yearname ('2001-10-19') returns '2001' with an underlying numeric value corresponding to '2001-01-01 00:00:00.000'

yearname ('2001-10-19', -1) returns '2000' with an underlying numeric value corresponding to '2000-01-01 00:00:00.000'

yearname ('2001-10-19', 0, 4) returns '2001-2002' with an underlying numeric value corresponding to '2001-04-01 00:00:00.000'

QuarterStart(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the first millisecond of the quarter containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the quarter which contains *date*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in Janu-

ary, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

quarterstart ('2005-10-29') returns '2005-10-01' with an underlying numeric value corresponding to '2005-10-01 00:00:00.000'

quarterstart ('2005-10-29', -1) returns '2005-07-01' with an underlying numeric value corresponding to '2005-07-01 00:00:00.000'

quarterstart ('2005-10-29', 0, 3) returns '2005-09-01' with an underlying numeric value corresponding to '2005-09-01 00:00:00.000'

QuarterEnd(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the last millisecond of the quarter containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the quarter which contains *date*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

quarterend ('2005-10-29') returns '2005-12-31' with an underlying numeric value corresponding to '2005-12-31 23:59:59.999'

quarterend ('2005-10-29', -1) returns '2005-09-30' with an underlying numeric value corresponding to '2005-09-30 23:59:59.999'

quarterend ('2005-10-29', 0, 3) returns '2005-11-30' with an underlying numeric value corresponding to '2005-11-30 23:59:59.999'

QuarterName(*date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a display value showing the months of the quarter (formatted according to the *MonthNames* script variable) and year with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the quarter. *Shift* is an integer, where the value 0 indicates the quarter which contains *date*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

quartername ('2005-10-29') returns 'Oct-Dec 2005' with an underlying numeric value corresponding to '2005-10-01 00:00:00.000'

quartername ('2005-10-29', -1) returns 'Jul-Sep 2005' with an underlying numeric value corresponding to '2005-07-01 00:00:00.000'

quartername ('2005-10-29', 0, 3) returns 'Sep-Nov 2005' with an underlying numeric value corresponding to '2005-09-01 00:00:00.000'

MonthStart(date [, shift = 0])

Returns a value corresponding to a timestamp with the first millisecond of the first date of the month containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the month which contains *date*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

monthstart ('2001-10-19') returns '2001-10-01' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

monthstart ('2001-10-19', -1) returns '2001-09-01' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

MonthEnd(date [, shift = 0])

Returns a value corresponding to a timestamp with the last millisecond of the last date of the month containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the month which contains *date*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

monthend ('2001-02-19') returns '2001-02-28' with an underlying numeric value corresponding to '2001-02-28 23:59:59.999'

monthend ('2001-02-19', -1) returns '2001-01-31' with an underlying numeric value corresponding to '2001-01-31 23:59:59.999'

MonthName(date [, shift = 0])

Returns a display value showing the month (formatted according to the `MonthNames` script variable) and year with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the month. *Shift* is an integer, where the value 0 indicates the month which contains *date*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

monthname ('2001-10-19') returns 'Oct 2001' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

monthname ('2001-10-19', -1) returns 'Sep 2001' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

MonthsStart(*n*, *date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the first millisecond of the *n* month period (starting from January 1st) containing *date*. The default output format will be the `DateFormat` set in the script. *N* must be (1), 2, (3), 4 or 6. *Shift* is an integer, where the value 0 indicates the period which contains *date*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

monthsstart (4, '2001-10-19') returns '2001-09-01' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

monthsstart (4, '2001-10-19', -1) returns '2001-05-01' with an underlying numeric value corresponding to '2001-05-01 00:00:00.000'

monthsstart (4, '2001-10-19', 0, 2) returns '2001-10-01' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

MonthsEnd(*n*, *date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a value corresponding to a timestamp with the last millisecond of the *n* month period (starting from January 1st) containing *date*. The default output format will be the `DateFormat` set in the script. *N* must be (1), 2, (3), 4 or 6. *Shift* is an integer, where the value 0 indicates the period which contains *date*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

monthsend (4, '2001-07-19') returns '2001-08-31' with an underlying numeric value corresponding to '2001-08-31 23:59:59.999'

monthsend (4, '2001-10-19', -1) returns '2001-08-31' with an underlying numeric value corresponding to '2001-08-31 23:59:59.999'

monthsend (4, '2001-10-19', 0, 2) returns '2002-01-31' with an underlying numeric value corresponding to '2002-01-31 23:59:59.999'

MonthsName(*n*, *date* [, *shift* = 0 [, *first_month_of_year* = 1]])

Returns a display value showing the months of the period (formatted according to the `MonthNames` script variable) and year with an underlying numeric value corresponding to a timestamp with the first millisecond of the *n* month period (starting from January 1st) containing *date*. *N* must be (1), 2, (3), 4 or 6. *Shift* is an integer, where the value 0 indicates the period which contains *date*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

monthsname (4, '2001-10-19') returns 'Sep-Dec 2001' with an underlying numeric value corresponding to '2001-09-01 00:00:00.000'

monthsname (4, '2001-10-19', -1) returns 'May-Aug 2001' with an underlying numeric value corresponding to '2001-05-01 00:00:00.000'

monthsname (4, '2001-10-19', 0, 2) returns 'Oct-Jan 2002' with an underlying numeric value corresponding to '2001-10-01 00:00:00.000'

WeekStart(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a value corresponding to a timestamp with the first millisecond of the first date (Monday) of the calendar week containing *date*. The default output format will be the `DateFormat` set in the script. *Shift* is an integer, where the value 0 indicates the week which contains *date*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

weekstart ('2006-01-12') returns '2006-01-09' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

weekstart ('2006-01-12', -1) returns '2006-01-02' with an underlying numeric value corresponding to '2006-01-02 00:00:00.000'

weekstart ('2006-01-12', 0, 1) returns '2006-01-10' with an underlying numeric value corresponding to '2006-01-10 00:00:00.000'

WeekEnd(*date* [, *shift* = 0 [,*weekoffset* = 0]])

Returns a value corresponding to a timestamp with the last millisecond of the last date (Sunday) of the calendar week containing *date*. The default output format will be the `DateFormat` set in the script.. *Shift* is an integer, where the value 0 indicates the week which contains *date*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

weekend ('2006-01-12') returns '2006-01-15' with an underlying numeric value corresponding to '2006-01-15 23:59:59.999'

weekend ('2006-01-12', -1) returns '2006-01-08' with an underlying numeric value corresponding to '2006-01-08 23:59:59.999'

weekend ('2006-01-12', 0, 1) returns '2006-01-16' with an underlying numeric value corresponding to '2006-01-16 23:59:59.999'

WeekName(*date* [, *shift* = 0 [,*weekoffset* = 0]])

Returns a display value showing the year and week number with an underlying numeric value corresponding to a timestamp with the first millisecond of the first date of the week containing *date*. *Shift* is an integer, where the value 0 indicates the week which contains *date*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

weekname ('2006-01-12') returns '2006/02' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

weekname ('2006-01-12', -1) returns '2006/01' with an underlying numeric value corresponding to '2006-01-02 00:00:00.000'

weekname ('2006-01-12', 0, 1) returns '2006/02' with an underlying numeric value corresponding to '2006-01-10 00:00:00.000'

LunarweekStart(*date* [, *shift* = 0 [,*weekoffset* = 0]])

Returns a value corresponding to a timestamp with the first millisecond of the lunar week (consecutive 7 day periods starting on Janu-

ary 1st each year) containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the lunar week which contains *date*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

lunarweekstart ('2006-01-12') returns '2006-01-08' with an underlying numeric value corresponding to '2006-01-08 00:00:00.000'

lunarweekstart ('2006-01-12', -1) returns '2006-01-01' with an underlying numeric value corresponding to '2006-01-01 00:00:00.000'

lunarweekstart ('2006-01-12', 0, 1) returns '2006-01-09' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

LunarweekEnd(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a value corresponding to a timestamp with the last millisecond of the lunar week (consecutive 7 day periods starting on January 1st each year) containing *date*. The default output format will be the *DateFormat* set in the script. *Shift* is an integer, where the value 0 indicates the lunar week which contains *date*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

lunarweekend ('2006-01-12') returns '2006-01-14' with an underlying numeric value corresponding to '2006-01-14 23:59:59.999'

lunarweekend ('2006-01-12', -1) returns '2006-01-07' with an underlying numeric value corresponding to '2006-01-07 23:59:59.999'

lunarweekend ('2006-01-12', 0, 1) returns '2006-01-15' with an underlying numeric value corresponding to '2006-01-15 23:59:59.999'

LunarWeekName(*date* [, *shift* = 0 [, *weekoffset* = 0]])

Returns a display value showing the year and week number with an underlying numeric value corresponding to a timestamp with the

first millisecond of the first date of the lunar week (consecutive 7 day periods starting on January 1st each year) containing *date*. *Shift* is an integer, where the value 0 indicates the lunar week which contains *date*. Negative values in *shift* indicate preceding lunar weeks and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in *weekoffset*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

lunarweekname ('2006-01-12') returns '2006/02' with an underlying numeric value corresponding to '2006-01-08 00:00:00.000'

lunarweekname ('2006-01-12', -1) returns '2006/01' with an underlying numeric value corresponding to '2006-01-01 00:00:00.000'

lunarweekname ('2006-01-12', 0, 1) returns '2006/02' with an underlying numeric value corresponding to '2006-01-09 00:00:00.000'

DayStart(*timestamp* [, *shift* = 0 [, *dayoffset* = 0]])

Returns a value corresponding to a timestamp with the first millisecond of the day contained in *timestamp*. The default output format will be the *TimestampFormat* set in the script. *Shift* is an integer, where the value 0 indicates the day which contains *date*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

Examples:

daystart ('2006-01-25 16:45') returns '2006-01-25 00:00:00' with an underlying numeric value corresponding to '2006-01-25 00:00:00.000'

daystart ('2006-01-25 16:45', -1) returns '2006-01-24 00:00:00' with an underlying numeric value corresponding to '2006-01-24 00:00:00.000'

daystart ('2006-01-25 16:45', 0, 0.5) returns '2006-01-25 12:00:00' with an underlying numeric value corresponding to '2006-01-25 12:00:00.000'

DayEnd(*timestamp* [, *shift* = 0 [, *dayoffset* = 0]])

Returns a value corresponding to a timestamp with the last millisecond of the day contained in *timestamp*. The default output format will be the *TimestampFormat* set in the script. *Shift* is an integer, where the value 0 indicates the day which contains *date*. Negative

values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

Examples:

dayend ('2006-01-25 16:45') returns '2006-01-25 23:59:59' with an underlying numeric value corresponding to '2006-01-25 23:59:59.999'

dayend ('2006-01-25 16:45', -1) returns '2006-01-24 23:59:59' with an underlying numeric value corresponding to '2006-01-24 23:59:59.999'

dayend ('2006-01-25 16:45', 0, 0.5) returns '2006-01-26 11:59:59' with an underlying numeric value corresponding to '2006-01-26 11:59:59.999'

DayName(*timestamp* [, *shift* = 0 [, *dayoffset* = 0]])

Returns a display value showing the date with an underlying numeric value corresponding to a timestamp with the first millisecond of the day containing *timestamp*. *Shift* is an integer, where the value 0 indicates the day which contains date. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

Examples:

dayname ('2006-01-25 16:45') returns '2006-01-25' with an underlying numeric value corresponding to '2006-01-25 00:00:00.000'

dayname ('2006-01-25 16:45', -1) returns '2006-01-24' with an underlying numeric value corresponding to '2006-01-24 00:00:00.000'

dayname ('2006-01-25 16:45', 0, 0.5) returns '2006-01-25' with an underlying numeric value corresponding to '2006-01-25 12:00:00.000'

timezone ()

returns the name of the current time zone as defined in Windows.

Example:

timezone ()

GMT () returns the current Greenwich Mean Time as derived from the system clock and Windows time settings.

Example:
gmt ()

UTC() returns the current Coordinated Universal Time.

Example:
utc()

daylightsaving ()

returns the current adjustment for daylight saving time as defined in Windows.

Example:
daylightsaving ()

setdateyear (*timestamp*, *year*)

returns a timestamp based on *timestamp* but with the year replaced with *year*.

timestamp is a standard QlikView timestamp (often just a date).

year is a four-digit year.

Examples:

setdateyear('2005-10-29', 2006) returns '2006-10-29'

setdateyear('2005-10-29 04:26', 2006) returns '2006-10-29 04:26'

setdateyearmonth (*timestamp*, *year*, *month*)

returns a timestamp based on *timestamp* but with the year replaced with *year*.

year is a four-digit year.

month is a one- or two-digit month.

Examples:

setdateyearmonth('2005-10-29', 2006, 3) returns '2006-03-29'

setdateyearmonth('2005-10-29 04:26', 2006, 3) returns '2006-03-29 04:26'

InYear (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the year containing *basedate*. The year can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the year which contains *basedate*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inyear ('2006-01-25', '2006-01-01', 0) returns true

inyear ('2005-01-25', '2006-01-01', 0) returns false

inyear ('2006-01-25', '2006-01-01', -1) returns false
inyear ('2005-01-25', '2006-01-01', -1) returns true
inyear ('2006-01-25', '2006-07-01', 0, 3) returns false
inyear ('2006-03-25', '2006-07-01', 0, 3) returns true

InYearToDate (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the part of year containing *basedate* up until and including the last millisecond of *basedate*. The year can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the year which contains *basedate*. Negative values in *shift* indicate preceding years and positive values indicate succeeding years. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inyeartodate ('2006-01-25', '2006-02-01', 0) returns true
inyeartodate ('2006-01-25', '2006-01-01', 0) returns false
inyeartodate ('2005-01-25', '2006-02-01', -1) returns true

InQuarter (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the quarter containing *basedate*. The quarter can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the quarter which contains *basedate*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inquarter ('2006-01-25', '2006-01-01', 0) returns true
inquarter ('2006-01-25', '2006-04-01', 0) returns false
inquarter ('2006-01-25', '2006-01-01', -1) returns false
inquarter ('2005-12-25', '2006-01-01', -1) returns true
inquarter ('2006-01-25', '2006-03-01', 0, 3) returns false
inquarter ('2006-03-25', '2006-03-01', 0, 3) returns true

InQuarterToDate (*date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the part of the quarter containing *basedate* up until and including the last millisecond of *basedate*. The quarter can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the quarter which contains *basedate*. Negative values in *shift* indicate preceding quarters and positive values indicate succeeding quarters. If you want to work with (fiscal) years not starting

in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inquartertodate ('2006-01-25', '2006-01-25', 0) returns true

inquartertodate ('2006-01-25', '2006-01-24', 0) returns false

inquartertodate ('2005-12-25', '2006-02-01', -1) returns false

InMonth (*date*, *basedate* , *shift*)

returns true if *date* lies inside the month containing *basedate*. The month can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the month which contains *basedate*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

inmonth ('2006-01-25', '2006-01-01', 0) returns true

inmonth ('2006-01-25', '2006-04-01', 0) returns false

inmonth ('2006-01-25', '2006-01-01', -1) returns false

inmonth ('2005-12-25', '2006-01-01', -1) returns true

InMonthToDate (*date*, *basedate* , *shift*)

returns true if *date* lies inside the part of month containing *basedate* up until and including the last millisecond of *basedate*. The month can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the month which contains *basedate*. Negative values in *shift* indicate preceding months and positive values indicate succeeding months.

Examples:

inmonthtodate ('2006-01-25', '2006-01-25', 0) returns true

inmonthtodate ('2006-01-25', '2006-01-24', 0) returns false

inmonthtodate ('2006-01-25', '2006-02-28', -1) returns true

InMonths (*n*, *date*, *basedate* , *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the *n* month period (aligned from January 1st) containing *basedate*. *N* must be (1), 2, (3), 4 or 6. The period can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the period which contains *basedate*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inmonths (4, '2006-01-25', '2006-01-01', 0) returns true

inmonths (4, '2006-01-25', '2006-05-01', 0) returns false
inmonths (4, '2006-01-25', '2006-01-01', -1) returns false
inmonths (4, '2005-12-25', '2006-01-01', -1) returns true
inmonths (4, '2006-01-25', '2006-03-01', 0, 3) returns false
inmonths (4, '2006-04-25', '2006-03-01', 0, 3) returns true

InMonthsToDate (*n*, *date*, *basedate*, *shift* [, *first_month_of_year* = 1])

returns true if *date* lies inside the part of the *n* month period (aligned from January 1st) containing *basedate* up until and including the last millisecond of *basedate*. *N* must be (1), 2, (3), 4 or 6. The period can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the period which contains *basedate*. Negative values in *shift* indicate preceding periods and positive values indicate succeeding periods. If you want to align with (fiscal) years not starting in January, you may indicate a value between 2 and 12 in *first_month_of_year*.

Examples:

inmonthstodate (4, '2006-01-25', '2006-04-25', 0) returns true
inmonthstodate (4, '2006-04-25', '2006-04-24', 0) returns false
inmonthstodate (4, '2005-11-25', '2006-02-01', -1) returns true

InWeek (*date*, *basedate*, *shift* [, *weekstart*])

returns true if *date* lies inside the week containing *basedate*. The week can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the week which contains *basedate*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in *weekstart*. This may be given as a real number indicating days and/or fractions of a day.

Examples:

inweek ('2006-01-12', '2006-01-14', 0) returns true
inweek ('2006-01-12', '2006-01-20', 0) returns false
inweek ('2006-01-12', '2006-01-14', -1) returns false
inweek ('2006-01-07', '2006-01-14', -1) returns true
inweek ('2006-01-12', '2006-01-09', 0, 3) returns false

InWeekToDate (*date*, *basedate*, *shift* [, *weekstart*])

returns true if *date* lies inside the part of week containing *basedate* up until and including the last millisecond of *basedate*. The week can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the week which contains *basedate*. Negative values in *shift* indicate preceding weeks and positive values indicate succeeding weeks. If

you want to work with weeks not starting midnight between Sunday and Monday, you may indicate an offset in days in `weekstart`. This may be given as a real number indicating days and/or fractions of a day.

Examples:

`inweektodate ('2006-01-12', '2006-01-12', 0)` returns true

`inweektodate ('2006-01-12', '2006-01-11', 0)` returns false

`inweektodate ('2006-01-12', '2006-01-05', -1)` returns true

`InLunarWeek (date, basedate , shift [, weekstart])`

returns true if `date` lies inside the lunar week (consecutive 7 day periods starting on January 1st each year) containing `basedate`. The lunar week can be offset by `shift`. `Shift` is an integer, where the value 0 indicates the lunar week which contains `basedate`. Negative values in `shift` indicate preceding lunar week s and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in `weekstart`. This may be given as a real number indicating days and/or fractions of a day.

Examples:

`inlunarweek ('2006-01-12', '2006-01-14', 0)` returns true

`inlunarweek ('2006-01-12', '2006-01-20', 0)` returns false

`inlunarweek ('2006-01-12', '2006-01-14', -1)` returns false

`inlunarweek ('2006-01-07', '2006-01-14', -1)` returns true

`inlunarweek ('2006-01-11', '2006-01-08', 0, 3)` returns false

`InLunarWeekToDate (date, basedate , shift [, weekstart])`

returns true if `date` lies inside the part of lunar week (consecutive 7 day periods starting on January 1st each year) containing `basedate` up until and including the last millisecond of `basedate`. The lunar week can be offset by `shift`. `Shift` is an integer, where the value 0 indicates the lunar week which contains `basedate`. Negative values in `shift` indicate preceding lunar week s and positive values indicate succeeding lunar weeks. If you want to work with an offset for the start of the lunar weeks, you may indicate an offset in days in `weekstart`. This may be given as a real number indicating days and/or fractions of a day.

Examples:

`inlunarweektodate ('2006-01-12', '2006-01-12', 0)` returns true

`inlunarweektodate ('2006-01-12', '2006-01-11', 0)` returns false

`inlunarweektodate ('2006-01-12', '2006-01-05', -1)` returns true

InDay (*timestamp*, *basetimestamp*, *shift* [, *daystart*])

returns true if *timestamp* lies inside the day containing *basetimestamp*. The day can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the day which contains *basetimestamp*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *daystart*, e.g 0.125 to denote 3am.

Examples:

inday ('2006-01-12 12:23', '2006-01-12 00:00', 0) returns true
inday ('2006-01-12 12:23', '2006-01-13 00:00', 0) returns false
inday ('2006-01-12 12:23', '2006-01-12 00:00', -1) returns false
inday ('2006-01-11 12:23', '2006-01-12 00:00', -1) returns true
inday ('2006-01-12 12:23', '2006-01-12 00:00', 0, 0.5) returns false
inday ('2006-01-12 11:23', '2006-01-12 00:00', 0, 0.5) returns true

InDayToTime (*timestamp*, *basetimestamp*, *shift* [, *daystart*])

returns true if *timestamp* lies inside the part of day containing *basetimestamp* up until and including the exact millisecond of *basetimestamp*. The day can be offset by *shift*. *Shift* is an integer, where the value 0 indicates the day which contains *basetimestamp*. Negative values in *shift* indicate preceding days and positive values indicate succeeding days. If you want to work with days not starting midnight, you may indicate an offset in fraction of a day in *daystart*, e.g 0.125 to denote 3am.

Examples:

indaytotime ('2006-01-12 12:23', '2006-01-12 23:59', 0) returns true
indaytotime ('2006-01-12 12:23', '2006-01-12 00:00', 0) returns false
indaytotime ('2006-01-11 12:23', '2006-01-12 23:59', -1) returns true

age(*timestamp*, *date_of_birth*)

Returns the age at the time of *timestamp* (in completed years) of a someone born on *date_of_birth*.

Example:

age('2007-01-25', '2005-10-29') returns 1
age('2007-10-29', '2005-10-29') returns 2

networkdays(*start_date*, *end_date* {, *holiday*})

Returns the number of working days (Monday-Friday) between and including *start_date* and *end_date* taking into account any optionally listed *holidays*. All parameters should be valid dates or timestamps.

Example:

networkdays ('2007-02-19', '2007-03-01') returns 9

networkdays ('2006-12-18', '2006-12-31', '2006-12-25', '2006-12-26') returns 8

firstworkdate(*end_date*, *no_of_workdays* {, *holiday*})

Returns the latest starting date to achieve *number_of_workdays* (Monday-Friday) ending no later than *end_date* taking into account any optionally listed holidays. *End_date* and *number_of_workdays* should be valid dates or timestamps.

Example:

firstworkdate ('2007-03-01', 9) returns '2007-02-19'

firstworkdate ('2006-12-31', 8, '2006-12-25', '2006-12-26') returns '2006-12-18'

lastworkdate(*start_date*, *no_of_workdays* {, *holiday*})

Returns the earliest ending date to achieve *number_of_workdays* (Monday-Friday) if starting at *start_date* taking into account any optionally listed holidays. *Start_date* and *number_of_workdays* should be valid dates or timestamps.

Example:

lastworkdate ('2007-02-19', 9) returns '2007-03-01'

lastworkdate ('2006-12-18', 8, '2006-12-25', '2006-12-26') returns '2006-12-29'

Number Interpretation Functions

The interpretation functions are a set of functions to interpret a field content or an expression. With these functions it is possible to state the data type, decimal separator, thousands separator etc. used.

If no interpretation functions are used, QlikView interprets the data as a mix of numbers, dates, times, time stamps and strings, using the default settings for number format, date format and time format defined by script variables and by the operating system.

In order to understand the examples in this section, you should first study the part “Number Format Dialogs” on page 301 in Book II.

Note For reasons of clarity, all number representations are given with decimal point as decimal separator.

num#(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **num#** function evaluates the *expression* numerically according to the string given as *format-code*. For a description of the format-code, see page 307 in Book II. Decimal separator and thousands separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the default number format set in the operating system is used.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
number format	# ##0,#	#,##0.#
num# (A, '#')	where A=35,648.375 returns	

Setting 1		Setting 2	
string	number	string	number
35,648.375	-	35648. 375	35648.375

num#(A, '#.#' , ',' , '.') where A=35,648.375 returns

Setting 1 and 2	
string	number
35,648.375	35648.375

num#(A, '#,#' , ',' , '.') where A=35648.375 returns

Setting 1 and 2		Note!
string	number	The number represent.!
35648.375	35648375	

num#(A, 'abc#,#') where A=abc123,4 returns

Setting 1		Setting 2	
string	number	string	number
abc123,4	123.4	abc123,4	1234

money#(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **money#** function evaluates the *expression* numerically according to the string given as *format-code*. For a description of the format-code, see page 307 in Book II. Decimal separator and

thousands separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the default number format set in the operating system is used.

The **money#** function generally behaves just like the **num#** function but takes its default values for decimal and thousand separator from the script variables for money format or the system settings for currency.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
money format	# ##0,00 kr	\$ #,##0.00
money# (A, '# ##0,00 kr')	where A=35 648,37 kr returns	

Setting 1		Setting 2	
string	number	string	number
35 648.37 kr	35648.37	35 648.37 kr	3564837

money#(A, '\$#', '.', ',') where A=\$35,648.37 returns

Setting 1 and 2	
string	number
\$35,648.37	35648.37

date#(*expression* [, *format-code*])

The **date#** function evaluates the *expression* as a date according to the string given as *format-code*. For a description of the format-code, see page 307 in Book II. If the format code is omitted, the default date format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
date# (A)	where A=8/6/97 returns	

Setting 1		Setting 2	
string	number	string	number
8/6/97	-	8/6/97	35648

date#(A, 'YYYY.MM.DD') where A=1997.08.06 returns

Setting 1 and 2	
string	number
1997.08.06	35648

time#(*expression* [, *format-code*])

The **time#** function evaluates the *expression* as a time according to the string given as *format-code*. For a description of the format-code, see page 307 in Book II. If the format code is omitted, the default time format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
time format	hh:mm:ss	hh.mm.ss
time#(A)	where A=09:00:00	returns

Setting 1		Setting 2	
string	number	string	number
09:00:00	0.375	09:00:00	-

time#(A, 'hh.mm') where A=09.00 returns

Setting 1 and 2	
string	number
09.00	0.375

timestamp#(*expression* [, *format-code*])

The **timestamp#** function evaluates the expression as a date and a time according to the string given as format-code. For a description of the format-code, see page 307 in Book II. If the format code is omitted, the default date and time formats set in the operating system are used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
time format	hh:mm:ss	hh:mm:ss
timestamp#(A)	where A=8/6/97 09:00:00	returns

Setting 1		Setting 2	
string	number	string	number
8/6/97 09:00:00	-	8/6/97 09:00:00	35648.375

timestamp#(A, 'YYYY-MM-DD hh_mm')

3.1415926535898	-
-----------------	---

Formatting Functions

The formatting functions are a set of functions to set the display format of a field or an expression. With these functions it is possible to set decimal separator, thousands separator etc. The easiest way to format numbers, times and dates is however in the number format dialog, see page 307 in Book II.

In order to understand the examples in this section, you should first study the chapter “Number Format Dialogs” on page 301 in Book II.

Note For reasons of clarity all number representations are given with decimal point as decimal separator.

num(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **num** function formats the expression numerically according to the string given as format-code. For a description of the format-code, see page 307 in Book II. Decimal separator and thousands separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the number format set in the operating system is used.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
number format	###0,#	#,##0.#
num (A , '0.0')	where A=35648.375	returns

Setting 1		Setting 2	
string	number	string	number
35 648 375	35648375	35648.375	35648.375

num(A , '#,##0.##' , ',' , '.') where A=35648 returns

Setting 1 and 2	
string	number
35,648.00	35648

num(pi() , '0,00') returns

Setting 1		Setting 2	
string	number	string	number
3,14	3.141592653	003	3.141592653

money(*expression* [, *format-code* [, *decimal-sep* [, *thousands-sep*]]])

The **money** function formats the *expression* numerically according to the string given as *format-code*. For a description of the format-code, see page 307 in Book II. Decimal separator and thousands separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the number format set in the operating system is used.

Examples:

The examples below assume the two following default settings:

	Default setting 1	Default setting 2
money format	kr # ##0,00	\$ #,##0.00
money (A)	where A=35648	returns

Setting 1		Setting 2	
string	number	string	number
kr 35 648,00	35648.00	\$ 35,648.00	35648.00

money(A , '#,##0 - ' , ' ' , ' ') where A=3564800 returns

Setting 1 and 2	
string	number
3,564,800 -	3564800

date(*expression* [, *format-code*])

The **date** function formats the expression as a date according to the string given as *format-code*. For a description of the format-code, see page 307 in Book II. If the format code is omitted, the date format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
date (A)	where A=35648	returns

Setting 1		Setting 2	
string	number	string	number
97-08-06	35648	8/6/97	35648

date(A, 'YY-MM-DD') where A=35648 returns

Setting 1 and 2	
string	number
97-08-06	35648

date(A, 'DD.MM.YYYY') where A=35648.375 return

Setting 1 and 2	
string	number
06.08.1997	35648.375

date(A, 'YY.MM.DD') where A=8/6/97 returns

Setting 1		Setting 2	
string	number	string	number
<NULL>	(nothing)	97.08.06	35648

time(*expression* [, *format-code*])

The **time** function formats the *expression* as a time according to the string given as *format-code*. For a description of the *format-code*, see page 307 in Book II. If the format code is omitted, the time format set in the operating system is used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
time format	hh:mm:ss	hh.mm.ss
time (A)	where A=0.375 returns	

Setting 1		Setting 2	
string	number	string	number
09:00:00	0.375	09.00.00	0.375

time(A) where A=35648.375 returns

Setting 1		Setting 2	
string	number	string	number
09:00:00	35648.375	09.00.00	35648.375

time(A, 'hh-mm') where A=0.99999 returns

Setting 1 and 2	
string	number
23-59	0.99999

timestamp(*expression* [, *format-code*])

The **timestamp** function formats the *expression* as a date and a time according to the string given as *format-code*. For a description of the *format-code*, see page 307 in Book II. If the format code is omitted, the date and time formats set in the operating system are used.

Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
date format	YY-MM-DD	M/D/YY
time format	hh:mm:ss	hh:mm:ss
timestamp(A)	where A=35648.375 returns	

Setting 1		Setting 2	
string	number	string	number
97-08-06 09:00:00	35648.37	8/6/97 09:00:00	35648.37
	5		5

timestamp(A, 'YYYY-MM-DD hh.mm')
where A=35648 returns

Setting 1 and 2	
string	number
1997-08-06 00.00	35648

interval(*expression* [, *format-code*])

The **interval** function formats the *expression* as a time interval according to the string given as a *format-code*. For a description of the *format-code*, see page 307 in Book II. If the format code is omitted, the time format set in the operating system is used.

Intervals may be formatted as a time, as days or as a combination of days, hours, minutes, seconds and fractions of seconds.

Examples:

The examples below assume the following operating system settings:

short date format	YY-MM-DD
time format	hh:mm:ss
number decimal format	.
interval(A)	where A=0.375 return

string	number
09:00:00	0.375

interval(A) where A=1.375 returns

string	number
33:00:00	1.375

interval(A, 'D hh:mm') where A=1.375 returns

string	number
1 09:00	1.375

interval(A-B, 'D hh:mm') where A=97-08-06 09:00:00 and B=96-08-06 00:00:00 returns

string	number
365 09:00	365.375

Chart Color Functions

These functions can be used in color expressions in sheet object properties supporting calculated color in image charts.

color(*n*)

Returns the color representation of color number *n* in the relevant chart color map. The color representation is a dual value where the text representation comes in the form of 'RGB(r, g, b)' where r, g and b are numbers between 0 and 255 representing the red, green and blue color value respectively. The number representation is an integer representing the red, green and blue components as defined in Visual Basic. Outside the calculated color expression in the **Color** page of the **Chart Properties** dialog, this function always returns black.

RGB(*e1, e2, e3*)

Returns the color representation of a color defined by the red component *e1*, the green component *e2* and the blue component *e3*. All three parameters must be expressions evaluating to integers in the range between 0 and 255. The color representation is a dual value where the text representation comes in the form of 'RGB(r, g, b)' where r, g and b are numbers between 0 and 255 representing the red, green and blue color value respectively. The number representation is an integer representing the red, green and blue components as defined in Visual Basic.

ARGB(*alpha, e1, e2, e3*)

Returns the color representation of a color defined by the red component *e1*, the green component *e2* and the blue component *e3* with an alpha-factor (opacity) of *alpha*. All four parameters must be expressions evaluating to integers in the range between 0 and 255. The color representation is a dual value where the text representation comes in the form of 'ARGB(a ,r, g, b)' where a, r, g and b are

numbers between 0 and 255 representing the alpha, red, green and blue color value respectively. The number representation is an integer representing the alpha, red, green and blue components as defined in Visual Basic.

colormix1(Value, ColorZero, ColorOne)

returns an RGB color representation from a gradient between two colors based on a value between 0 and 1. If the value is equal to zero the first color will be returned. If it is 1, the second color is returned and if it is in between the appropriate intermediate shading is returned.

Value is a real number between 0 and 1.

ColorZero is a valid RGB color representation for the color to be associated with the low end of the interval.

ColorOne is a valid RGB color representation for the color to be associated with the high end of the interval.

Example:

colormix1(x, black (), red())

colormix2(Value, ColorMinusOne, ColorOne [,ColorZero])

returns an RGB color representation from a gradient between two colors based on a value between -1 and 1 with the possibility to specify an intermediate color for the center position. If the value is equal to -1 the first color will be returned. If it is 1, the second color is returned and if it is in between the appropriate intermediate shading is returned.

Value is a real number between -1 and 1.

ColorMinusOne is a valid RGB color representation for the color to be associated with the low end of the interval.

ColorOne is a valid RGB color representation for the color to be associated with the high end of the interval.

ColorZero is a optional valid RGB color representation for the color to be associated with the center of the interval.

Examples:

colormix2(x, red (), green()) returns colors from red to green via brown

colormix2(x, red (), green(), black()) returns colors from red to green via black

Note The RGB, HSL and SYSCOLOR functions always return a color with the alpha value of 255 (opaque).

HSL (*hue, saturation, luminosity*)

Returns the color representation of a color defined by a *hue* value between 0 and 1, a *saturation* value between 0 and 1 and a *luminosity* value between 0 and 1. The color representation is a dual value where the text representation comes in the form of 'RGB(r, g, b)' where r, g and b are numbers between 0 and 255 representing the red, green and blue color value respectively. The number representation is an integer representing the red, green and blue components as defined in Visual Basic.

black(*[alpha]*)

Returns the RGB color representation for black (RGB 0,0,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

darkgray(*[alpha]*)

Returns the RGB color representation for dark gray (RGB 128,128,128). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

lightgray(*[alpha]*)

Returns the RGB color representation for light gray (RGB 192,192,192). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

white(*[alpha]*)

Returns the RGB color representation for white (RGB 255,255,255). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

blue(*[alpha]*)

Returns the RGB color representation for blue (RGB 0,0,128). Optionally a parameter for alpha-factor can be given. An *alpha* of 0

corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

lightblue(*[alpha]*)

Returns the RGB color representation for light blue (RGB 0,0,255). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

green(*[alpha]*)

Returns the RGB color representation for green (RGB 0,128,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

lightgreen(*[alpha]*)

Returns the RGB color representation for light green (RGB 0,255,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

cyan(*[alpha]*)

Returns the RGB color representation for cyan (RGB 0,128,128). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

lightcyan(*[alpha]*)

Returns the RGB color representation for light cyan (RGB 0,255,255). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

red(*[alpha]*)

Returns the RGB color representation for red (RGB 128,0,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

lightred(*[alpha]*)

Returns the RGB color representation for light red (RGB 255,0,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0

corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

magenta(*[alpha]*)

Returns the RGB color representation for magenta (RGB 128,0,128). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

lightmagenta(*[alpha]*)

Returns the RGB color representation for light magenta (RGB 255,0,255). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

brown(*[alpha]*)

Returns the RGB color representation for brown (RGB 128,128,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

yellow(*[alpha]*)

Returns the RGB color representation for yellow (RGB 255,255,0). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

qliktechblue(*[alpha]*)

Returns the RGB color representation for qliktech blue (RGB 8,18,90). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

qliktechgray(*[alpha]*)

Returns the RGB color representation for qliktech gray (RGB 158,148,137). Optionally a parameter for alpha-factor can be given. An *alpha* of 0 corresponds to full transparency. An *alpha* of 255 corresponds to full opacity.

syscolor(*nr*)

Returns the RGB color representation for the Windows system color *nr*, where *nr* corresponds to the parameter to the Windows API function `GetSysColor(nr)`. Some values for *nr* are:

0	COLOR_SCROLLBAR
1	COLOR_BACKGROUND
2	COLOR_ACTIVECAPTION
3	COLOR_INACTIVECAPTION
4	COLOR_MENU
5	COLOR_WINDOW
6	COLOR_WINDOWFRAME
7	COLOR_MENUTEXT
8	COLOR_WINDOWTEXT
9	COLOR_CAPTIONTEXT
10	COLOR_ACTIVEBORDER
11	COLOR_INACTIVEBORDER
12	COLOR_APPWORKSPACE
13	COLOR_HIGHLIGHT
14	COLOR_HIGHLIGHTTEXT
15	COLOR_BTNFACE
16	COLOR_BTNSHADOW
17	COLOR_GRAYTEXT
18	COLOR_BTNTEXT
19	COLOR_INACTIVECAPTIONTEXT
20	COLOR_BTNHIGHLIGHT
21	COLOR_3DDKSHADOW
22	COLOR_3DLIGHT
23	COLOR_INFOTEXT
24	COLOR_INFOBK
26	COLOR_HOTLIGHT (Win2000)
27	COLOR_GRADIENTACTIVECAPTION (Win2000)
28	COLOR_GRADIENTINACTIVECAPTION (Win2000)

20 EXAMPLES

This chapter contains further examples illustrating functions and expressions in QlikView.

20.1 Examples of Aggregate Qualifiers

These examples are made with the **sum** function, but could be applied on all chart aggregation functions supporting set definitions and the **total** qualifier.

Example 1:

Study the following table when no selections are made:

Month	Grp	sum(Val)	sum(total Val)	sum({1} total Val)
		21	21	21
1	A	1	21	21
1	B	2	21	21
2	A	3	21	21
2	B	4	21	21
3	A	5	21	21
3	B	6	21	21

The second and third expression columns will have the same number in all rows. This number equals the calculated total in the first expression column.

Now let's select only Month 1 and 2. The result will look as follows:

Month	Grp	sum(Val)	sum(total Val)	sum({1} total Val)
		10	10	21
1	A	1	10	21
1	B	2	10	21
2	A	3	10	21
2	B	4	10	21

The result of the fifth column remains unchanged, since the set definition disregards current selections. The second expression with the **total** qualifier will display the new total of 10, which is still equal to the total in the first expression (third column).

Example 2:

Study the following table:

Month	Grp	sum(Val)	sum(total Val)	sum(total<Month> Val)	sum(total<Grp> Val)
		21	21	21	21
1	A	1	21	3	9
1	B	2	21	3	12
2	A	3	21	7	9
2	B	4	21	7	12
3	A	5	21	11	9
3	B	6	21	11	12

In the third expression column $sum(total<Month> Val)$ one total is calculated for each Month.

In the fourth expression column $sum(total<Grp> Val)$ one total is calculated for each Grp.

20.2 Examples of Ranking Functions

The examples below are made with the **rank** (**vrank**) function but could be applied in a similar manner to the **hrank** function. It should however be noted that the **hrank** function is only relevant in pivot tables.

Example 1:

Study the two single-dimension straight tables below:

Month	sum(Val)	rank(sum(Val))
	110	-
1	17	3
2	9	7
3	22	1
4	16	4
5	10	6
6	5	8
7	11	5
8	20	2

Month	sum(Val)	rank(sum(Val))
	110	-
3	22	1
8	20	2
1	17	3
4	16	4
7	11	5
5	10	6
2	9	7
6	5	8

The two tables are the same but the left table is sorted by the first column while the right table is sorted by the last column. This example shows the basic functionality of **rank**. The highest value has the highest rank (lowest ranking number).

The **rank** functions always return NULL in total rows.

Example 2:

Study the two-dimensional pivot table below:

Group	Month	sum(Val)	rank(sum(Val))	rank(total sum(Val))
A	1	17	2	3
	4	16	3	4
	7	11	4	5
	8	20	1	2
	Total	64	-	-
B	2	9	3	7
	3	22	1	1
	5	10	2	6
	6	5	4	8
	Total	46	-	-
Total		110	-	-

This table is based on the same data as the two tables in the first example. You can now see how the current column segment is limited to rows with the same value in the *Group* column in the multi-dimensional case. Months within group A are ranked separately from months in group B. By introducing the **total** qualifier, an overall ranking can again be achieved.

Example 3:

This example will demonstrate the effect of the different modes for the numeric representation of the ranking. Study the table below:

Month	sum(X)	rank(sum(X))	mode=0	mode=1	mode=2	mode=3	mode=4
	86	-	-	-	-	-	-
4	20	1	1	1	1	1	1
2	12	2-3	2	2	2.5	3	2
7	12	2-3	2	2	2.5	3	3
3	10	4-5	4.5	4	4.5	5	5
5	10	4-5	4.5	4	4.5	5	4
8	9	6	6	6	6	6	6
6	7	7	7	7	7	7	7
1	6	8	8	8	8	8	8

The third column shows the ranking in text representations while columns 4-8 shows the number representation of the same ranking in different modes. The expression in each column is

$$\text{num}(\text{rank}(\text{sum}(X), \text{mode}))$$

where *mode* is 0 to 4.

Mode 0 (default)

Rows 2 and 3 share ranking but are clearly on the lower half of the total ranking. Their number representation therefore is rounded downwards to 2. Rows 4 and 5 share ranking but fall just over the middle of the ranking table and therefore get a number representation of the average of the first and last rank in the column ($(1+8)/2 = 4.5$). This mode is especially useful when you want to use visual cues to mark the highest and lowest ranking data within a group.

Mode 1

In both cases the lower ranking figure within the group is used, which is 2 for rows 2 and 3, and 4 for rows 4 and 5.

Mode 2

In both cases the average of the low and high ranking within the group is used, which is 2.5 ($(2+3)/2$) for rows 2 and 3, and 4.5 ($(4+5)/2$) for rows 4 and 5.

Mode 3

In both cases the higher ranking figure within the group is used, which is 3 for rows 2 and 3, and 5 for rows 4 and 5.

Mode 4

Each row gets its own distinct numeric value. The order within the groups sharing a ranking is determined by the sort order of the chart's dimensions.

Example 4:

This example will demonstrate the effect of the different formats for the text representation of the ranking function. Study the table below:

Month	sum(x)	rank(sum(x),0,0)	rank(sum(x),0,1)	rank(sum(x),0,2)
	86	-	-	-
4	20	1	1	1
2	12	2-3	2	2
7	12	2-3	2	
3	10	4-5	4	
5	10	4-5	4	4
8	9	6	6	6
6	7	7	7	7
1	6	8	8	8

Columns 3-5 show the text representation of the same ranking function with different values in the format parameter.

Format 0 (default)

Rows sharing ranking are shown as 'low value – high value', e.g. '2-3' and '4-5'.

Format 1

Rows sharing ranking always get the number of the lowest rank as text representation, in this case e.g. 2 for rows 2 and 3.

Format 2

One row in each group sharing the same ranking gets the low rank number as text representation, while the other rows within the group get a blank string. The order within the groups sharing a ranking is determined by the sort order of the charts dimensions.

20.3 Examples of Chart Inter-Record Functions

Top Function

The examples below are made with the **top** function but could be applied in a similar manner to the **bottom**, **first** and **last** functions. It should however be noted that the **first** and **last** functions are only relevant in pivot tables.

Example 1:

Study the single-dimension straight table below:

Month	sum(Val)	top(sum(Val))	sum(Val)/top(sum(Val))
	21	3	700%
1	3	3	100%
2	7	3	233%
3	11	3	367%

In the single-dimension case the **top** function will always refer to the first data row of the table (the total row is not counted).

Note that expressions using the **top** function will be properly evaluated also in the total row since the total has a clear relation to a specific column segment, in this case the entire column.

Example 2:

Study the table below. It is a two-dimensional straight table sorted primarily on Grp:

Month	Grp	sum(Val)	top(sum(Val))	top(total sum(Val))
		21	-	1
1	A	1	1	1
2	A	3	1	1
3	A	5	1	1
1	B	2	2	1
2	B	4	2	1
3	B	6	2	1

The **top** function without the **total** qualifier will now return the expression evaluated on the top row within the innermost sort group (*Grp* dimension in this case). One value will be returned for *Grp*=A and one for *Grp*=B.

By using the **total** qualifier in a multi-dimensional case you may again refer to the absolute top row of the table with the same value being returned for all rows.

The expression using the **top** function without the **total** qualifier will evaluate to NULL in the total row, since it cannot be clearly associated with a specific column segment.

The expression using the **top** function with the **total** qualifier will of course be evaluated for the column segment spanning the entire column.

Now let's convert the same table to a pivot table with all totals turned on (below):

Month	Grp	sum(Val)	top(sum(Val))	top(total sum(Val))
1	A	1	1	1
	B	2	1	1
	Total	3	1	-
2	A	3	3	1
	B	4	3	1
	Total	7	3	-
3	A	5	5	1
	B	6	5	1
	Total	11	5	-
Total		21	-	1

The expression using the **top** function without the **total** qualifier will evaluate to NULL in the grand total row, since it cannot be clearly associated with

a specific column segment. However all the partial sums will be evaluated for each column segment.

The expression using the **top** function with the **total** qualifier will lack values in the partial totals but will return a value in the grand total row.

Example 3:

Study the table below:

Two-dimension straight table - sort on Grp				
Month	Grp	sum(Val)	top(sum(Val))	sum(Val)/top(sum(Val))
		21	-	-
1	A	1	1	100%
2	A	3	1	300%
3	A	5	1	500%
1	B	2	2	100%
2	B	4	2	200%
3	B	6	2	300%

Finally we may change the inter-field sort order so that the chart is sorted primarily on *Month*. The original table would then look as follows:

Two-dimension straight table - sort on Month				
Month	Grp	sum(Val)	top(sum(Val))	sum(Val)/top(sum(Val))
		21	-	-
1	A	1	1	100%
1	B	2	1	200%
2	A	3	3	100%
2	B	4	3	133%
3	A	5	5	100%
3	B	6	5	120%

Above Function

The examples below are made with the **above** function but could be applied in a similar manner to the **below**, **before** and **after** functions. It should however be noted that the **before** and **after** functions are only relevant in pivot tables.

Example 1:

Study the single-dimensional straight table below:

Single-dimension straight table - above			
Month	sum(Val)	above(sum(Val))	sum(Val)/above(sum(Val))
	21	-	-
1	3	-	-
2	7	3	233%
3	11	7	157%

The third column shows the expression `sum(Val)` evaluated one row above the current row, which can be confirmed by comparing with the values for `sum(Val)` in the second column. On the first row the **above** function will return NULL, as there is no row above on which to evaluate the expression. The **above** function always returns NULL on all total rows.

The fourth column demonstrates the most typical use of this function, i.e. to calculate the difference between e.g. different time periods.

Example 2:

Study the two-dimensional pivot table below:

Two-dimension pivot table - effect of total qualifier				
Grp	Month	sum(Val)	above(sum(Val))	above(total sum(Val))
A	1	1	-	-
	2	3	1	1
	3	5	3	3
	Total	9	-	-
B	1	2	-	5
	2	4	2	2
	3	6	4	4
	Total	12	-	-
Total		21	-	-

The **above** function without the **total** qualifier (fourth column) will only act within each sort group. A NULL value will be returned on the top row of each column segment.

When adding a **total** qualifier (fifth column), the entire column will be regarded as one column segment. Only the very top row will return NULL. All total rows are disregarded and return NULL in themselves.

RowNo and NoOfRows

The example below is made with the **RowNo** and **NoOfRows** functions but could be applied in a similar manner to the **ColumnNo** and **NoOfColumns** functions. It should however be noted that the **ColumnNo** and **NoOfColumns** functions are only relevant in pivot tables.

Example 1:

Study the two-dimensional pivot table below:

rowno() and noofrows()						
Month	Grp	rowno()	rowno(total)	noofrows()	noofrows(total)	
1	A	1	1	2	6	
	B	2	2	2	6	
	Total	0	-	2	-	
2	A	1	3	2	6	
	B	2	4	2	6	
	Total	0	-	2	-	
3	A	1	5	2	6	
	B	2	6	2	6	
	Total	0	-	2	-	
Total		-	0	-	6	

Column 3

The **RowNo** function without the **total** qualifier will return the row number within each sort group column segment. In subtotals rows the row number 0 will be returned, since these totals clearly belong to a specific column segment. In the grand total row NULL will be returned.

Column 4

The **RowNo** function with the **total** qualifier will return the row number within the entire column. In subtotals rows a NULL value will be returned. In the grand total row 0 will be returned.

Column 5

The **NoOfRows** function without the **total** qualifier will return the number of data rows within each sort group column segment. In subtotal rows the same number as in the data rows above will be returned. In the grand total row NULL will be returned.

Column 6

The **NoOfRows** function with the **total** qualifier will return the number of data rows within the entire column, which is also what will be returned in the grand total row. In subtotal rows NULL will be returned.



21 NESTED AGGREGATIONS AND RELATED ISSUES

This chapter exemplifies some important techniques in relation to nested aggregations and the use of **aggr** function in charts.

21.1 Nested Aggregations with total Qualifier

As a general rule, it is not allowed to nest aggregations in a QlikView chart expression. From version 7.5 there is however one very important exception to this rule. As long as you use the **total** qualifier in the inner aggregation function, the nesting is allowed.

Say for example that you want to calculate the sum of the field *Sales*, but only include transactions with and *OrderDate* equal to the last year. The last year can be obtained via the aggregation function **max(total year(OrderDate))**.

An aggregation as follows would then do the job:

```
sum( if(year(OrderDate)=max(total year(OrderDate)), Sales).
```

The inclusion of the **total** qualifier is absolutely necessary for this kind of nesting to be accepted by QlikView, but then again also necessary for the desired comparison. The need for this type of nesting is quite common and should be used wherever suitable.

21.2 Nested Aggregations with the aggr Function

Nesting with **total** is not always enough. For more generic nesting capabilities you will have to use the **aggr** function in combination with calculated dimensions. Let us look at a small example:

The following data has been read from the script:

Original data from script	
SalesRep	Customer
Donna Brown	Bechtel Corporation
Karl Anderson	Berkeley Design
Donna Brown	Capitolnet Marketing Group (CMG)
Karl Anderson	Chas T. Main, Inc.
Karl Anderson	Degolyer and MacNaughton
Lisa Taylor	ediSys
John Doe	Fimetrics Systems
Kathy Clinton	HCS
Lisa Taylor	Homestead Custom
Lisa Taylor	Illuminati
John Doe	Metro-Goldwyn-Mayer, Inc.
Lisa Taylor	Onetouch Interactive
Peggie Hurt	Savetz Publishing
William Fisher	TECC
William Fisher	VA Research
Lisa Taylor	XYZ Operations

Figure 92. The data from the script

An obvious question given this data would be: “How many customers does each sales rep have?”. This is easily done in a standard chart:

First question: How many customer does each sales rep have?	
SalesRep	count(Customer)
Donna Brown	2
John Doe	2
Karl Anderson	3
Kathy Clinton	1
Lisa Taylor	5
Peggie Hurt	1
William Fisher	2

Figure 93. The first order of aggregation

Now however, let us ask a couple of new questions based on the knowledge just gained: “How many sales reps only have one single customer? “How many have three or more?”. If we disregard the fact that you in this simple case can count the numbers in the expression columns by hand, these are types of questions that require a second order of aggregation. The data necessary to make the calculation does not exist in the original fields, nor can it be directly calculated from them.

We must simply find a way to use the expression column in the chart above as a dimension in a new chart. The answer lies in the **aggr** function. By stating

`=aggr(count(Customer),SalesRep)`

as dimension we can essentially perform the calculation of the first chart as an 'inner chart calculation' in a new chart. The new chart could then be given the expression

`count(distinct SalesRep)`

and the trick is complete. The distinct qualifier is necessary, since QlikView will count the number of lines in the underlying table. The resulting table will look like below:

2nd question: How many salesreps have 1, 2, 3 etc customers?	
<code>=aggr(count(Customer),SalesRep)</code>	<code>count(distinct SalesRep)</code>
1	2
2	3
3	1
5	1

Figure 94. The second order of aggregation

Two things to be noted:

The second chart does in no way require the presence of the first chart. It is fully self-contained with the first order aggregation defined within its dimension.

The possibilities of nesting do not end here. The dimension arguments of the **aggr** function may of course contain calculated dimensions, which in turn make use of the **aggr** function. It should however be relatively easy to loose track of what you are doing when passing the third level of aggregation.

21.3 Sum of rows in Pivot Tables

The QlikView straight table has a choice for its totals between a simple sum of rows and a calculated expression total. The QlikView pivot table lacks this choice. Pivot table totals are always calculated as expression total.

This is normally a good thing, since it is a rather rare occasion that a sum of rows total is relevant when the two differ. You should exercise extreme care when using sum of rows on any type of aggregation other than pure sums.

Let's say that we have a school contest where three person teams get points by their grades in three different classes. The team may select the highest score within a group for each individual class and then add the three top scores together for a total. The following data has been read from the script:

Original data from script		
Class	Name	Score
English	John	5
English	Karen	1
English	Lisa	4
History	John	3
History	Karen	3
History	Lisa	2
Math	John	3
Math	Karen	3
Math	Lisa	4

Figure 95. The data from the script

We must now make a chart with *Class* as dimension and **max(Score)** as expression. A straight table with sum of rows will look as follows:

Straight table with sum of rows	
Class	max(Score)
History	3
Math	4
English	5
	12

Figure 96. Sum of rows in straight table

If we for some reason want to display this in a pivot table (not much use here, but if we had more dimensions it may make sense), we run into problems. The straight table above converted into a pivot table would look like as follows:

Pivot table with expression total	
Class	max(Score)
English	5
History	3
Math	4
Total	5

Figure 97. Expression total in pivot table

In this specific case the total of 12 is clearly what we want and 5 is equally wrong for our purposes. Again the **aggr** function comes to our rescue. In this case we use it in the expression, not in the dimension.

The original expression is enclosed in an **aggr** function, using the surrounding chart's dimension also as dimension in the **aggr** function. Then we use this bundle as argument to a **sum** aggregation. The result will look as follows:

Pivot table with sum of rows	
Class	sum(aggr(max(Score),Class))
English	5
History	3
Math	4
Total	12

Figure 98. Sum of rows in pivot table

As you see the total is again the one desired. What happened?

Well the beauty of the **aggr** function is that in the individual rows it will evaluate to only a single value. This is because the dimension obviously only has one possible value on each ordinary data row. Since the inner dimension and expression are the same as for the surrounding chart, each value will of course be exactly the same as the result without the enclosing **sum** and **aggr** functions.

For the total row however, the **aggr** function will return three values, one for each value of the dimension field. These will in turn be summed by the **sum** aggregation, while formally still being an expression total, the result equals that of sum of rows.

21.4 Linear Regression in Table Charts

Linear regression trend lines can be shown in QlikView bitmap charts by means of the **Trendlines** option in the **Expressions** page of the **Chart Properties**. It is also possible to display the regression equation.

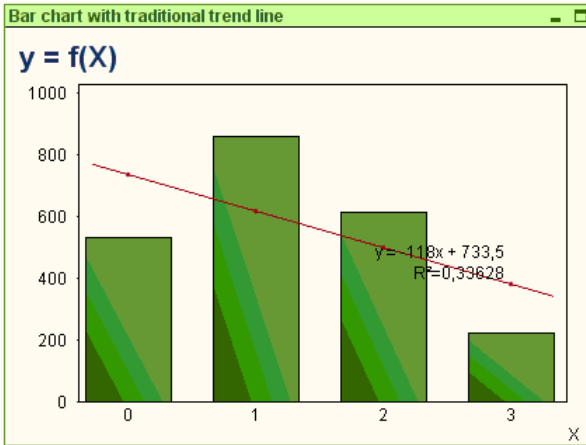


Figure 99. A bar chart with a traditional trendline

However, if you want to display the regression data in e.g. a table chart, the regression must be calculated. The **linest_m** and **linest_b** aggregation functions will give you the required slope and y-intercept values of the linear regression.

To calculate correctly, these functions need to have the entire chart aggregation (expression iterated over dimension) as input. This can be achieved by defining an **aggr** function containing the same base expression and dimension(s) as the containing chart. The **aggr** function is then used as parameters to the **linest** aggregations. The resulting expression could look like follows:

linest_m(total aggr(Y,X),X)*X + linest_b(total aggr(Y,X),X)

The **only** function is implied around all occurrences of X and Y. The **linest** aggregations should be made with the **total** qualifier, else would the regression parameters be calculated per data point rather than for the whole set of data. The result can be seen in the combo chart below.

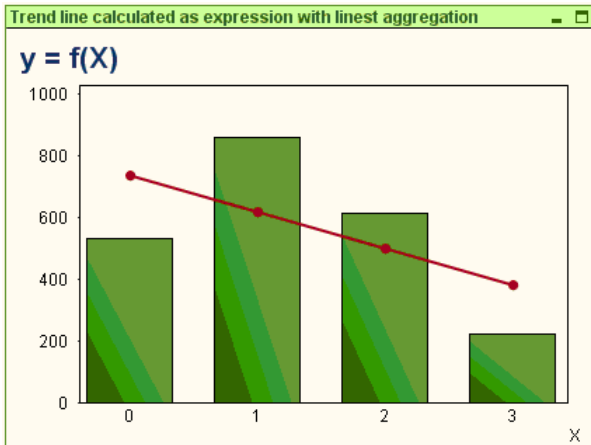


Figure 100. The same data as combo chart with the regression as a regular line expression

Note that the trend line here is not a traditional QlikView trend line, but a regular expression plotted as a line. You can see the difference from the fact that the expression plot, as opposed to a traditional trend line, is not extrapolated outside the first and last data points.

This chart can be converted into a straight table where the regression values are shown in cells.

Trend line calculated as expression with linest aggregation						
X	Y	linreg y=mx+b	R2	m	b	
0	533	733,5	0,33628	-118,0	733,5	
1	859	615,5	0,33628	-118,0	733,5	
2	612	497,5	0,33628	-118,0	733,5	
3	222	379,5	0,33628	-118,0	733,5	

Figure 101. The calculated regression values in straight table format

In the straight table above three extra columns have been added to show the m, b and R² values. These of course are constant for all table rows. The expressions needed would like follows, in order of appearance:

linest_r2(total aggr(Y,X),X)

linest_m(total aggr(Y,X),X)
linest_b(total aggr(Y,X),X)

22 CALCULATED FORMULAS

In the property dialogs of the QlikView sheets and sheet objects, there are a number of properties allowing fixed text labels or fixed numbers. These are typically used as labels, window titles, chart titles and in some cases as fixed numeric limits.

For many of the property entries mentioned above, it is possible to enter a calculated expression rather than a constant text or number. This feature is called calculated formula. Wherever a calculated formula can be used, this is indicated on the relevant place in this manual.

22.1 Entering a Calculated Formula

Calculated formulas are entered according to the following syntax:

= expression

For the syntax of allowed expressions, see the section below.

The equal sign in the first position of the entry indicates that the remainder should be interpreted as an expression. QlikView will try to evaluate the expression. If this is not possible, e.g. because of incorrect syntax, the entire label including the equal sign will be displayed.

Calculated formulas can also be generated in the **Edit Expression** dialog which opens when clicking on the ... button beside the edit box (see further on page 249).

22.2 Expression Syntax for Calculated Formulas

The syntax for *expression* in calculated labels is virtually equal to that of chart expressions (see page 293). There are however a few exceptions:

- As there are no dimensions to iterate over, the aggregation functions will essentially behave as if used in chart expressions with the **total** qualifier before all field names. The **total** qualifier is thus optional and has no special meaning in calculated formulas.
- In a calculated formula, field names may optionally be used without an enclosing aggregation function. In that case, **only** will be used as aggregation function.

Example:

```
= Currency  
is equal to  
= only(Currency)
```

22.3 Error Messages

If a calculated formula cannot be correctly evaluated by QlikView, the formula itself will be returned, followed by two slashes and an error message.

Example:

```
= mode(x) //out of object memory
```

Each calculated formula requires a certain amount of memory. In order not to use excessive memory, a limit on the allowed memory allocation for each calculated label has been put in to the program. If you enter a too complex expression, QlikView will return the expression followed by the error message “// out of object memory”. You can solve this problem by increasing the memory allocated (see page 125 in Book I).


23 FIELD GROUPS


One main difference between QlikView and many other database viewers, OLAP tools etc. is that in QlikView there is no need to predefine any hierarchies in the input data. The unique internal logic of QlikView gives you the complete freedom to access any field as a full dimension in any order you like. For most purposes this freedom is extremely powerful.

However, there are occasions when a predefined hierarchy could actually help you to display data more efficiently. QlikView therefore offers the possibility to define groups of fields. The groups can be hierarchic (drill-down) or non-hierarchic (cyclic).

Groups are created in the **Groups** page of the **Document Properties** dialog. They can be used in charts, where they appear together with the available fields in the dimension drop-down boxes on the **Dimension** (see page 31).

Any fields can be grouped together.

Drill-down groups are marked with an  icon,

while cyclic groups are displayed with an  icon.

23.1 Hierarchic Groups (Drill-Down)

When several fields form a natural hierarchy, it makes sense to create a drill-down group. Typical examples of hierarchic groups could be:

Time: Year, Quarter, Month

or

Geography: Continent, Country, State, City

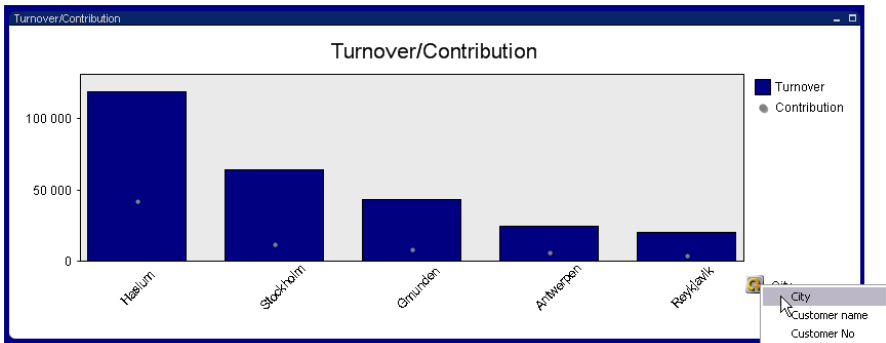
When a drill-down group is used as a dimension in a chart, the chart will use the first field in the group's list of fields that has more than one possible value. If selections are made that cause the field to have only one possible value, the next field in the list will be used instead, provided that it has more than one possible value. If no field in the list has more than one possible value, the last field will be used anyway.

In the first example above, Year will be used as chart dimension until a single year is selected. The chart will then show Quarter. If a single quarter is selected, the chart will switch to Month.

As selections disappear, so that more than one value becomes possible in the upper fields of the group's field list, the chart will automatically be drilled back up. Forced drill-up can be achieved by clicking on the drill-up icon in the chart.



23.2 Non-Hierarchic Groups (Cyclic)



Sometimes it may be useful to group also fields which do not form a natural hierarchy or even have nothing in common at all. The reason would be to enable the user to make quick changes of the data to be displayed in a chart.

Any fields can be grouped together in a cyclic group. When a cyclic group is used as a dimension in a chart, the chart will initially use the first field in the group's list of fields. The user may then switch to another field by clicking the cycle icon in the chart. The fields are used in the order they appear in the group's field list. When the last field in the list has been used, the turn goes back to the first field. The chart may be cycled indefinitely.

It is also possible to right-click on the cycle icon, in which case a list of the fields in the cyclic group is displayed for direct selection (see the picture above).

Cyclic groups should not be confused with cyclic display of expressions in charts.

24 CUSTOM ERROR MESSAGES

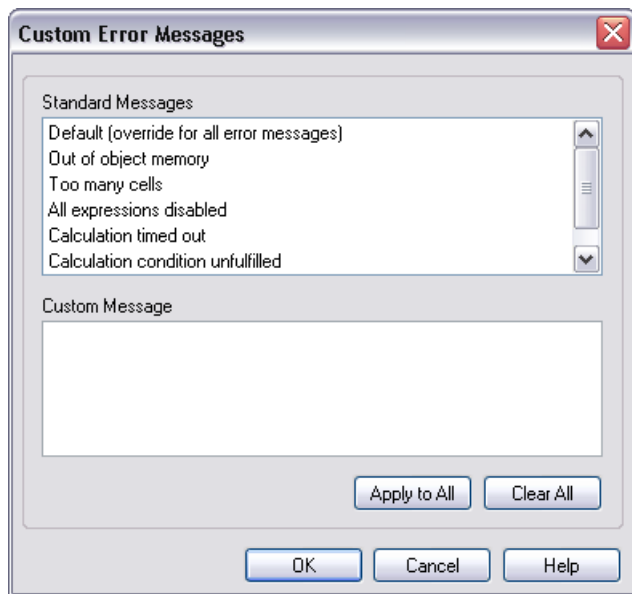


Figure 102. The Custom Error Messages dialog.

QlikView offers the possibility of customizing error messages in charts and table boxes. The **Custom Error Message** dialog is opened via the **Error Messages** button, which is found on the **General** page of the **Chart Properties** dialog, as well as on the **General** page of the **Table Box Properties** dialog. If you wish all the messages to be the same, select **Default (override for all error messages)**, then type your text.

Standard Messages

List of the standard error messages. To customize a message, select it and type a text of your choice in the **Custom Message** text box.

Custom Message

Here you enter the text to be shown instead of the standard message selected above. The text may be a calculated formula.

Apply to All Objects

Click this button to apply the custom messages to all the calculated objects of the document

Clear All

Click this button to clear all the custom error messages.

Help

Provides help on this dialog.

GLOSSARY



GLOSSARY

absolute path	The location of a file, folder or directory on a disk, as seen from the root or the top level. See also relative path.
active	A window, dialog box, tabbed sheet or list box that is currently in use. Also called current.
access restriction	Method to restrict access of a QlikView document for different users or user groups
aggr function	The Aggr() function is a function that is used for nested aggregations. Its first parameter must be an aggregation function – the inner aggregation function. The result is a set of records that in turn can be aggregated by an outer aggregation function. Example: Sum(Aggr(Count(...), ...)) .
aggregation function	An aggregation function is a function that operates on a set of values – several records - to return a single scalar value. Examples: Sum() , Count() , Avg() . Aggregation functions must be used in charts. They can also be used in the script when the “group by” clause is used. See also Range function and Scalar function.
AJAX	Shorthand for "Asynchronous JavaScript and XML", is a development technique for creating interactive web applications. See also QlikView AJAX zero footprint client.
alert	An alert is an entity that can send or show a warning message if a specific condition is fulfilled, e.g. when something in the data is different from what it should be.
alternative value	A field value that is excluded by a selection in the same list box, but not by selections made in other list boxes. The cell is gray by default, but can be made to be colored white, just like optional cells, by enabling Show Alternatives in the List Box Properties dialog.

application	Software designed to carry out a specific kind of activity, such as user access in a database. An application could be a combination of a program and a document. In this manual application usually means a program, such as Excel or QlikView, but could sometimes also denote a specific solution programmed as a QlikView document.
bookmark	A saved set of selections which can be recalled by a user and shared with other users.
bookmark object	A sheet object where bookmarks can be created, deleted and recalled.
calculated dimension	A dimension where the set of values are not defined by the values of a field, but rather by the possible values of an expression.
chart	A common name for bar charts, pie charts, funnel charts, line charts, grid charts, scatter plots, pivot tables and straight tables. Charts show aggregated values of the data, i.e. one item in a chart (a cell in a pivot table, a bar in a bar chart, etc.) corresponds to several records in the logical tables.
circular reference	A structure in the data model where the keys between at least three tables form a circle. The logic then inferred is usually ambiguous and the circular reference needs to be broken. QlikView does this by setting one of the tables as loosely coupled.
clear selections	An operation that resets a QlikView application to a state where no selections are made, i.e. no field values are excluded.
color coding	The state of a field value is shown by the color of its cell.
column	Term commonly used instead of field in databases.
concatenate (1)	An operation that uses two tables and combines them into one. The two tables are merely added to each other, i.e. data are not changed and the produced table contains the same number of records as the original tables together. Several concatenate operations can be performed sequentially, so that the produced table is the concatenation of several tables.

concatenate (2)	An operation that uses two text strings and combines them into one. The operator used for this is the ampersand “&”.
cross table	A table with two or more dimensions, where some dimensions are vertical and some are horizontal. QlikView can use cross tables as input tables. QlikView can also display pivot tables as cross tables.
crosstable	A QlikView script prefix used to load cross tables.
current selections box	A sheet object that shows selections in the fields and their logical status.
custom object	A placeholder sheet object for custom OCX controls in QlikView controls.
csv	File extension for comma separated value files. Used when a table is stored as a text file.
DBMS	DataBase Management System. The database program used in the storage and modification of data in the database.
data source	A name for a physical database, defined through the ODBC interface.
delimiter	A character or code that marks the beginning or end of an item, such as a sentence, paragraph, page, record, field or word. This word is sometimes used instead of ‘separator’.
dialog box	A window that lets you select options and activate those options by choosing the appropriate command buttons. Some dialog boxes display warnings and messages you need before taking action.
dimension	A chart dimension is the set of values for the chart to iterate over when it calculates the values for its expression(s). In the simple case one could say that it is what appears on the x-axis in a standard bar chart. Normally a dimension consists of a field but it may also be a group or a calculated expression.
document	A QlikView file saved in binary format having the default extension qvw. Also called QlikView file.

driver	A program that runs in the background and takes care of the communication with a peripheral device such as a printer or monitor, or with another program.
dual fields	Fields that have both textual and numeric representations, e.g. dates, months, formatted numbers, etc.
edit module dialog	The text editor where VB script macros are created and tested.
edit script dialog	The text editor where the load script is edited.
excluded	Field value that cannot be chosen or selected without changing one or more previous selections. The cell is colored gray to show its status.
expression	A formula. Expressions can be used in a number of places in QlikView: In the load script, in charts, as dynamic labels, in text boxes, etc.
field	Equivalent to column in database. In QlikView, a field is typically represented by a list box (see also value and list box).
font	The typesetting for letters and characters. Fonts are described by name, appearance and size, as in "Arial bold 10pt".
forced exclusion	Also called not selection. Can only be made in and list boxes. It is made by keeping the mouse button depressed until the cell turns red.
format painter	The paint brush on the tool bar. It can be used to transfer properties from one sheet object to another.
generic database	A generic database is a table in which the field names are stored as field values in one column, while the field values are stored in a second. Generic databases are usually used for attributes of different objects. To load and transform generic databases in QlikView, use the generic prefix.
help, html help	The normal Windows help that is invoked from the Help menu or by hitting the F1 key.
hidden script	A part of the load script that can be hidden and password protected.

in-memory analysis	Term for BI tools that hold the entire data set in primary memory and calculate all necessary aggregations on demand.
input box	A sheet object that is used for entering data into QlikView variables and displaying their values.
input field	A field that is declared as an Input field can be used for input. Good for planning, forecasting and budgeting.
intervalmatch	A QlikView script prefix used to match discrete values to intervals.
join	An operation that uses two tables and combines them into one. The records of the table produced are combinations of records in the two original tables, usually such that the two records contributing to any given combination in the produced table have a common value for one or several common fields, a so called natural join. In QlikView, joins can be made in the script, producing logical tables. Inner join, Left join, Right join and Outer join are all possible operations.
keep	A script operation that uses two tables. Just as a join, the keep operation will keep the values that have matches in the other table, but it will not merge the two tables into one. Inner keep, Left keep and Right keep are all possible operations
key field	A field that exists in two tables and connects the two.
line/arrow object	A sheet object used for adding lines or arrows to the layout.
list box	A sheet object that displays a list of field values.
load	The script statement used to load data from files or from select statements. Load statements are evaluated by QlikView, as opposed to Select statements that are evaluated by the ODBC driver or the OLE DB provider.
log file	A file that, like a log book, logs everything that happens during a script execution.
logical table	Tables with data that QlikView evaluates. The logical tables are the tables produced when the script is executed.

mapping	An operation for data cleansing in the script. Mapping is based on mapping tables that are loaded by means of the mapping prefix.
multi box	A two-column sheet object, in which each row in the first column contains a field name, and the second contains a drop-down with the corresponding field values.
nested aggregation	A calculation using a two-step aggregation. See aggr function.
OCX replacement control	A windowless OCX control which is incorporated into the QlikView layout via a custom object. OCX replacement controls can be programmed by the user or by third parties. See also custom object.
OLE DB	Object Linking and Embedding for Databases. A way for applications to communicate with databases. Different types of data sources can be read via this interface, notably ODBC data sources.
ODBC	Open DataBase Connectivity. A way for applications to communicate with databases. An ODBC driver is one or several system dynamic-link libraries (DLL) that allow ODBC-enabled programs such as QlikView to access specified data sources and retrieve data created in another format, such as dBASE.
optional	A field value that can be chosen or selected without changing the status of any previous selection. The cell is colored white to show its status, just like alternative cells.
path	The location of a file, folder or directory on a disk. See absolute/relative path.
pivot table	A sheet object that allows several dimensions, several expressions with aggregated data, pivoting and grouping.
primary memory	A computer's internal memory, which is made up of RAM.
QlikView AccessPoint	Single point of access to corporate QlikView documents. Requires QlikView Publisher.
QlikView AJAX zero footprint client	A lightweight object-based client to the QlikView Server that is based on the AJAX technology.

QlikView Java client	Java client for access to QlikView documents published by QlikView Server.
QlikView Java Objects client	Object-based Java client for access to QlikView documents published by QlikView Server. Provides better design possibilities than QlikView Java client.
QlikView OCX	QlikView packaged as an ActiveX component for integration into third party software. Not to be confounded with OCX replacement controls.
QlikView Plug-In	Short for QlikView Analyzer for Microsoft Internet Explorer.
QlikView Publisher	Administrative tool for automated reload and publishing of QlikView documents. Can be used as stand-alone or together with one or more of the other components in the QlikView Administrator suite.
QlikView Server	Software for publishing QlikView documents to online users using a client-server architecture.
QlikX	Object-based QlikView OCX for integration into other products or into web sites.
qvd file	A native QlikView file format. A qvd file contains one data table, no layout and no security. It is basically a “binary csv file”, optimized for fast loading.
RAM	Abbreviation of Random Access Memory. Usually synonymous to primary memory.
range function	A range function is a function that operates on a set of values – several parameters - to return a single scalar value. See also Aggregation function and Scalar function.
record	Equivalent to a row in a table.
relative path	The location of a file, folder or directory on a disk, as seen from a specified directory, usually the directory of the QlikView document. See also absolute path.
reload	The QlikView script needs to be run in order to refresh data in the QlikView application. This can be done manually or automatically using a scheduled task or using the QlikView Publisher.

resident	A table that has been loaded in the script can be accessed using a Load ... resident statement in the script.
report	With “report” one usually refers to a static paper or pdf report. A QlikView document is in that sense in itself not a report.
report editor	QlikView has a built-in report editor in which one can prepare standard paper reports for printing.
reserved fields	Fields used in the access restriction management: <i>USERID</i> , <i>PASSWORD</i> , <i>SERIAL ACCESS</i> , <i>NTNAME</i> , <i>NTDOMAINSID</i> , <i>NTSID</i> and <i>OMIT</i> .
RTF format	Rich Text Format. A method of encoding formatted text for easy transfer between applications. A file saved in RTF format keeps attributes like font, style etc.
scalar function	A scalar function is a function that operates on a single value to return a single scalar value. Examples: <code>chr()</code> , <code>sin()</code> , <code>applymap()</code> . See also Aggregation function and Range function.
script	A description of what data to load. The script is a small program that is run by QlikView. When executed, it connects your QlikView document to one or several data sources or opens text files and reads the specified information into QlikView. The data source and the fields to be included are defined here.
section access	A section of the QlikView load script that can be used for access restriction.
select	Selection of field values by clicking them is the very essence of QlikView. Select is also a SQL statement used in most query tools.
semantic links	Selections can be made indirectly through semantic links. These are similar to field values, but they describe the relations between the objects rather than the objects themselves. Semantic links are stored in semantic tables which are loaded by means of the semantic prefix.
separator	A character or code that separates one column or field from the next. Sometimes referred to as delimiter.

sheet	A QlikView screen initially contains an empty sheet with a tab (name tag) attached to it. Different objects, such as list boxes or charts, can be put on the sheet, and several sheets can be created in a document.
slider/calender object	A sheet object which can be used to select values in a field or set values to one or two variables.
SQL	Structured Query Language. A standard for making queries in relational databases.
start page	When starting QlikView, you will as the first window see a start page where you can access examples, favorites and recently used documents and connections.
statement	A script command can also be called statement. All statements must end with semicolons “;”.
statistics box	A sheet object that displays a set of statistical entities, calculated on the possible field values of the corresponding field.
stored procedures	Programs that are stored in, and run on Databases.
straight table	A sheet object that allows several dimensions, several expressions with aggregated data and have good sorting capabilities.
synthetic keys	Keys between tables that are generated internally by QlikView in cases where there are several keys linking the tables. Synthetic keys are sometimes an indication of a poorly or even incorrectly designed data model.
system fields	Fields generated by QlikView. The system fields contain information on the origin of the fields that have been read into QlikView. This information can be used to identify the files or tables of the data source where a specific field name is to be found.
system variables	Parameters generated by QlikView in the script for special purposes, e.g. definitions of document number formats
tabbed sheet	See sheet.
tab in script	The script can be split up onto several tabs. This improves the structure and facilitates navigation in the script.

table box	A sheet object that contains a record oriented view of data. Any set of fields can be chosen as columns, and possible combinations of field values are shown on different lines.
theme	A file containing the properties and look and feel of a document, sheet or object. Themes can be created and used from inside QlikView.
URL	Universal Resource Locator. A general address of the type used on the World Wide Web.
variable	A named entity which can be given one single value. Variables can be used in the script and in expressions in charts and other sheet objects.
X64	Computer architecture for 64-bit technology.
XML	Extended Markup Language. More complex than html, but not as complex as SGML. One XML file can contain one or several tables.

APPENDIX

- **Limitations and Requirements**
- **Keyboard Command Shortcuts**
- **Frequently Asked Questions**
- **Data Protection**
- **Sharing Documents**
- **Backus-Naur Formalism (Syntax Notation)**



A LIMITATIONS AND REQUIREMENTS

A.1 Limitations of the Amount of Data to be Loaded

The amount of data that can be loaded into a QlikView document is very large. It is primarily limited by the amount of primary memory of the computer. However, there are some inherent limitations in QlikView that one must be aware of when building very large documents.

A QlikView document cannot have more than:

Number of fields	only limited by RAM
Number of tables	only limited by RAM
Number of distinct values in one field	only limited by RAM

Each table, loaded through one script statement, or concatenated through several script statements, cannot have more than:

Number of cells	only limited by RAM
Number of rows	only limited by RAM

A.2 System Requirements

This version of QlikView is designed to run on an IBM PC compatible computer running Windows™. This section describes the hardware and software requirements to run QlikView on your computer.

Hardware

You need a computer with

- a Pentium II processor or higher. (Pentium ®4 or better recommended.) QlikView makes full use of modern multi-core processor architecture and we thus strongly recommend using such processors.
- graphics hardware supporting color XGA resolution, or better.
- a monitor with a resolution of at least 1024 x 768.
- a mouse or an equivalent pointing device supported by MS Windows.
- a CD-ROM drive (if installation is to be made from CD).
- a hard disk with at least 150 MB of free disk space.
- enough primary memory (see below).

Operating System

You need one of the following operating systems:

- Microsoft® Windows Server 2003™; or
- Microsoft® Windows XP™ or
- Windows Vista;

Additional Requirements for QlikView x64

- Processor supporting the x64 architecture
- 4 GB RAM
- Windows XP Professional x64 Edition™; or
- Windows Server 2003 x64 Edition™

Note QlikView 9 does not ship in a native IA-64 (Itanium) version. The 32-bit version will run on Itanium based machines with Windows, but performance can be expected to be poor. For native 64-bit Itanium support we refer you to use QlikView version 7.xx.

Note The installation requires the presence of Microsoft® MDAC 2.5 or later. This is automatically installed with e.g. Microsoft Office and all versions of Microsoft Internet Explorer 5.5 and later

Memory

Primary Memory

Although it is possible to run QlikView with less memory, the use of a computer with at least 1 GB for 32-bit Windows and 4 GB for 64-bit Windows is recommended.

The amount of memory you need for QlikView depends on the amount of data you load, and on the data structure. Too little primary memory will drastically affect the performance of QlikView in a negative way.

If you notice that your computer accesses the hard disk every time you make a selection in QlikView, you either have too many programs running, or too little primary memory.

Virtual Memory

Make sure that your swap file (the virtual memory) is large enough. The swap file should in principle be as large as possible. However,

if you have limited disk space, or a lot of primary memory, you may want a smaller swap file or no swap file.

The system usually takes care of the swap file size dynamically. However, if you have too little free space on the hard disk, you may get problems with your swap file getting too small. In such a case, you can fix the size of the swap file. This is done in the **Virtual memory** dialog, which is accessed from the task bar: choose **Settings, Control Panel, System, Performance, Virtual Memory**.

Note 32-bit Windows sets a limit of 2 GB logical address space for any one program running (3 GB for Windows Advanced Server). QlikView can never make use of more memory than this, regardless of the size of physical memory in the computer. This limitation does not exist when running QlikView 64-bit edition.

Note Defragment the hard disk before changing the swap file settings. It usually leads to a larger contiguous free hard disk area and thus a larger possible swap file.

A.3 Dynamic Link Libraries (dll's) Needed

In the windows system directories of all the clients there must be a number of dll's. They are all installed during the installation procedure. For information on the dll's required, see the ReadMe file in your QlikView directory. QlikView uses Windows Installer and msi technology for installation, which ensures that the correct versions of system dll's are always used.



B KEYBOARD COMMAND SHORTCUTS

In this chapter you will find a list of the different keyboard command shortcuts available in QlikView.

B.1 File Menu Command Shortcuts

CTRL+N

Equivalent to **New** on the **File** menu.



CTRL+O

Equivalent to **Open...** on the **File** menu.



CTRL+SHIFT+O

Equivalent to **Open in Server...** on the **File** menu.

CTRL+S

Equivalent to **Save** on the **File** menu.



F12

Equivalent to **Save As...** on the **File** menu.

CTRL+P

Equivalent to **Print...** on the **File** menu.



CTRL+SHIFT+P

Equivalent to **Print As PDF...** on the **File** menu.



CTRL+E

Equivalent to **Edit Script...** on the **File** menu.



CTRL+R

Equivalent to **Reload** on the **File** menu.



CTRL+SHIFT+R

Equivalent to **Partial Reload** on the **File** menu.

CTRL+T

Equivalent to **Table Viewer...** on the **File** menu.



B.2 Edit Menu Command Shortcuts

CTRL+Z

Equivalent to **Undo Layout Change** on the **Edit** menu.



CTRL+Y

Equivalent to **Redo Layout Change** on the **Edit** menu.



CTRL+X

Equivalent to **Cut** on the **Edit** menu.



CTRL+C

Equivalent to **Copy** on the **Edit** menu.



CTRL+V

Equivalent to **Paste** on the **Edit** menu.



DEL Equivalent to **Remove** on the **Edit** menu.

CTRL+A

Equivalent to **Activate All** on the **Edit** menu.

CTRL+F

Equivalent to **Search** on the **Edit** menu.



SHIFT+CTRL+F

Equivalent to **Advanced Search** on the **Edit** menu.

B.3 View Menu Command Shortcuts

CTRL+Q

Equivalent to **Current Selections...** on the **View** menu.



CTRL+G

Toggles layout design grid on or off.

B.4 Selections Menu Commands Shortcuts

SHIFT+LEFTARROW

Equivalent to **Back** on the **Selections** menu.



SHIFT+RIGHTARROW

Equivalent to **Forward** on the **Selections** menu.



CTRL+SHIFT+L

Equivalent to **Lock** on the **Selections** menu (locks all selections).



CTRL+SHIFT+U

Equivalent to **Unlock** on the **Selections** menu (unlocks all selections).



CTRL+SHIFT+D

Equivalent to **Clear** on the **Selections** menu (clears all selections).



B.5 Settings Menu Command Shortcuts

CTRL+ALT+U

Equivalent to **User Preferences...** on the **Settings** menu.



CTRL+ALT+D

Equivalent to **Document Properties...** on the **Settings** menu.



CTRL+ALT+S

Equivalent to **Sheet Properties...** on the **Settings** menu.



CTRL+ALT+V

Equivalent to **Variable Overview...** on the **Settings** menu.



CTRL+ALT+E

Equivalent to **Expression Overview...** on the **Settings** menu



B.6 Bookmarks Menu Command Shortcuts

CTRL+B

Equivalent to **Add Bookmark...** on the **Bookmarks** menu.



CTRL+SHIFT+B

Equivalent to **More...** on the **Bookmarks** menu.



B.7 Tools Menu Command Shortcuts

CTRL+M

Equivalent to **Edit Module...** on the **Tools** menu.



CTRL+ALT+A

Equivalent to **Alerts...** on the **Tools** menu.



B.8 Object Menu Command Shortcuts (List box, Statistics box and Open Multi Box)

CTRL+L

Equivalent to **Lock** on the **Object** menu (locks selections in active object).

CTRL+U

Equivalent to **Unlock** on the **Object** menu (unlocks selections in active object).

CTRL+D

Equivalent to **Clear** on the **Object** menu (clears selections in active object).

ALT+ENTER

Equivalent to **Properties...** on the **Object** menu (opens the **Properties** dialog in active object).

B.9 Script Keyboard Shortcuts

To see a list of the available keyboard shortcuts in the **Script** type CTRL+QSC in the script pane.

CTRL+G

Go to line number in the script.

CTRL+K,C

Comment lines in script.

CTRL+K,U

Uncomment lines in script.

CTRL+Q,T,A

Add tab in script.

CTRL+Q,T,P

Promote active tab.

CTRL+Q,T,D

Demote active tab.

CTRL+Q,T,N

Rename active tab.

CTRL+Q,T,R

Remove active tab.

CTRL+Q,Q

Creates an autogenerated script.

CTRL+Q,U,I

Opens **Unicode input utility**.

CTRL+Q,J,P,G

Creates a script for reading attributes from jpeg files.

CTRL+Q,M,P,3

Creates a script for reading attributes from mp3 files.

CTRL+Q,W,M,A

Creates a script for reading attributes from wma files.

B.10 F Key Keyboard Shortcuts

F1 Activates context sensitive help.

F3 Enters search mode if a searchable object is activated.

F6 Activates the sheet tab to the left of the currently active tab.

F7 Activates the sheet tab to the right of the currently active tab.

F12 Equivalent to **Save As...** on the **File** menu.

CTRL+F6
Activates the leftmost sheet tab.

CTRL+F7
Activates the rightmost sheet tab.

C APPLICATION PERFORMANCE OPTIMIZATION

C.1 Introduction

With small or medium sized QlikView applications you generally don't have to worry too much about the design of the application for reasons of performance. As the amount of data grows, both time and memory constraints may become very evident if the application is poorly designed. If you are lucky, some simple design alterations can give you very substantial gains in performance. This appendix points out a few common pit falls and suggests remedies for them.

In general performance is improved by moving the “problem“ from application objects to the script driven database. This is often a trade off situation. Response time is enhanced and ad hoc capability is diminished. The recommendations below should not be seen as universally beneficial. Use them when they improve the general state of the application or when they make that little bit of difference that makes or breaks.

The following is a list of examples of applied methods for the handling of the problems above. They are meant to illustrate the problem and point at useful QlikView functionality. It is not possible to give a general recommendation as to which method is best, but the order of the examples is an indication.

C.2 Count (Distinct 'FieldName')

Count(distinct fieldname) is generally speaking a performance consuming operation, especially for test fields. When possible, replace the *count()* and the distinct qualifier with *sum()* by assigning the value '1' to each distinct occurrence as it is read in the script.

The script could look as follows:

```
Load
    Alfa,
    if (peek('Alfa')=Alfa,0,1) as Flag1,
    Num
resident table_1
order by Alfa Asc;
```

Here the “peek“ compares the value of Alfa being read with that previously read. If the values are the same, “Flag“ is set to 0. If they are different, “Flag” is set to 1. The number of distinct values will then be equal to sum(Flag). Please note that the list has to be ordered and that when using “order by“ in a load resident QlikView orders the list before starting to read.

Another method:

```
Load distinct
Alfa,
Alfa as AlfaDist
resident table_1;
```

Now Count(DistinctAlfa) can be replaced by a simple count: Count(AlfaDist). Notice that Alfa is read twice, once with the original name to link to the original table, once with a new name to allow Count(). (Linking fields is not allowed in Count()). All other fields must also be left out as they would degrade the distinct clause. A third method is to give each distinct value of “Alfa“ a numeric value:

table_2:

```
Load
Alfa,
Autonumber(Alfa) as AlfaNum,
Num
resident table_1;
```

Count(Distinct AlfaNum) is a cheaper operation than Count(Distinct Alfa) since the comparison is of numeric values. An ever cheaper method is to find the last (or largest) result of the autonumber function.

```
set AlfaDistinctCount = peek( 'AlfaNum', -1, 'table_2'
);
```

in the script or as expression:

```
max( AlfaNum)
```

in a layout object.

C.3 If (Condition(Text),....)

If clauses involving text comparisons is generally expensive. Solutions can be to map text to numbers e.g. by using *autonumber* (see examples in the previous section) and/or do the test in the script.

The testing of text strings is slower than numeric testing. Consider the expression

```
If (Alfa= 'ABC', 'ABC', left (Alfa, 2))
```

The test could be done directly in the script without losing any flexibility.

```
Load
*,
  If (Alfa = 'ABC', 1, 0) as Flag
resident table_1 ;
```

The expression becomes

```
If ( Flag = 1, 'ABC', left (Alfa, 2))
```

and the test is much simpler.

C.4 Sum (If (Condition, 'FieldName'...))

Here the aggregation is independent of the table dimensions and the result is distributed over the dimensions of the table. The problem can be treated either by doing the test in the script and aggregating in the table or by doing the whole operation in the script. There are numerous techniques for this e.g. *interval match*, *group by*, *peek*, *if...then...else*.

This case involves two steps namely the testing of “Condition“ and the aggregation of the result. If we take the previous example and add the aggregation

```
Sum ( If (Alfa= 'ABC', Num*1.25 , Num) )
```

```
Load
*,
  If (Alfa = 'ABC', 1, 0) as Flag
resident table_1 ;
```

The expression becomes

```
Sum ( If ( Flag = 1, Num* 1.25 , Num ) )
```

The aggregation can also be done directly in the script as follows:

```

table_2:
Load
*,
    If (Alfa = 'ABC', 1, 0) as Flag
resident table_1 ;

table_3:
Load
    Alfa,
    If ( Flag = 1, Num* 1.25 , Num ) as NewNum
resident table_2 ;

table_4:
Load
    Alfa,
    Sum( NewNum ) as SumNum
resident table_3
group by Alfa ;

```

Note The aggregation is done over Alfa as this is the dimension in the test.

C.5 If (Condition, Sum('FieldName')..)

This construction is included here only to emphasize the difference to the previous case. This aggregation is completely contextual and generally speaking does not cause performance problems.

C.6 If (Condition1, Sum('FieldName'), If (Condition2, Sum('FieldName').....

The logic of nested *If...then else...* is conceptually easy but can often become troublesome to administer. We have seen cases with hundreds of nesting levels. This is both memory as well as CPU intensive. The “Conditions“ can often be replaced by transforming them. A typical example is aggregating *quantity*price* where price is variable. This can be handled by “extended interval match“. If two conditions, e.g.

“A AND B”

are to be satisfied the test might be replaced by a condition “C“.

Example:

```
sum((GAC12_STD_COST * GAC15_EXCHANGE_RATE) *
GIV24_DISP_QTY)
```

Replaces

```
Sum(
  If((GAC12_EFCT_DT<= GIV23_REJ_DT and
      GAC12_EXPIRE_DT>GIV23_REJ_DT) and
      (GAC15_EFCT_DT<= GIV23_REJ_DT and
      GAC15_EXPIRE_DT>GIV23_REJ_DT),
      GAC12_STD_COST * GAC15_EXCHANGE_RATE) *
      GIV24_DISP_QTY,
  Null())
and

  Sum(
    If(GAC12_EFCT_DT<= GIV23_REJ_DT,
    If(GAC12_EXPIRE_DT>GIV23_REJ_DT,
    If(GAC15_EFCT_DT<= GIV23_REJ_DT,
    If(GAC15_EXPIRE_DT>GIV23_REJ_DT,
    (GAC12_STD_COST * GAC15_EXCHANGE_RATE) *
    GIV24_DISP_QTY,
    Null()))))
```

by reading the fields `GAC12_STD_COST` and `GAC15_EXCHANGE_RATE` as slowly changing dimensions. (Please see page page 479 in Book I).

C.7 Sorting text

QlikView automatically evaluates if a *Field* is to be treated as *numeric*, *text* or *general*. Fields evaluated as *text* will be sorted as *text* which is the slowest sort operation. This can be replaced manually to sort by load order. If sorting of list boxes etc is not needed, turn it off.

C.8 Dynamic captions and text objects

Dynamically calculated expressions can be entered almost anywhere where you can enter text. The resources required for evaluation of an expression is however dependent on its environment. Expressions in charts and tables that are defined in the expressions dialog are only calculated when the object is visible and data changes. They are e.g. not calculated when the object is minimized.

On the other hand, if the object title is calculated this calculation is performed every time any change occurs. There are also numerous ways of defining show conditions, calculation conditions etc. These tests will also be performed at all times.

Some expressions are more expensive than others and become even more expensive the more frequently they have to be evaluated. The introduction of asynchronous calculation has shifted the behavior and maybe these effects have become more noticeable in your applications.

The time functions e.g. **Now()** and **Today()** will be evaluated whenever a recalculation is required. Especially the **Now()** function can become quite costly since it causes a recalculation of the application every second.

For example:

```
    If ( ReloadTime()+3>Now(), 'Old Data', 'New Data')
```

Here one might consider

```
    If ( ReloadTime()+3>Today(), 'Old Data', 'New Data')
```

As a simple test, put the expressions into text boxes. Then try sizing the text box with **Now()** in it.

C.9 Macro triggers (“on change“)

Macros can be set to be triggered by almost any event taking place in the application. Beware of cascading or recursive events where one event triggers the next which in turn.

D FREQUENTLY ASKED QUESTIONS

D.1 Installation

- Q: What are the requirements for installing QlikView?
- A: The QlikView installation package uses Windows installer technique and you must have version 2.0 or higher running on your system. If the package isn't deployed by a system administrator with a deployment tool, the account that is to run the installation must have administrator privileges.
- Q: What is Windows installer?
- A: The Windows installer is standard developed by Microsoft for software installation, modification, repair and removal. The installation package, also known as the msi, comprises of a database and the files to be installed.
- Q: Why does QlikView use msi and Windows installer for installation?
- A: The Windows installer and msi packages guarantees that the program is installed in a way that does not create conflicts with other programs or with Windows. It also facilitates correct uninstall, should you want to remove QlikView at a later point in time. Windows Installer and msi packages is the installation method recommended by Microsoft and is required for all certification of software products with Windows.
- Q: How do I determine what version of Windows installer I have?
- A: In the "Windows\system32" folder find the file called "msi.dll", open properties and go to the "version" page.
- Q: I have an old version of Windows installer, where can I obtain the latest?
- A: The latest version can be found on Microsoft's home page.

-
- Q: Can the installation overwrite important system files?
- A: No, the overwriting of files is handled by the Windows installer service itself which uses the Windows File Protection rules. To read more about these rules, see Microsoft's home page.
- Q: Where can I find more information about Windows installer?
- A: There are several good home pages on the Internet about Windows installer. Here are a few:
- msdn.microsoft.com
 - www.appdeploy.com
 - www.wise.com

D.2 QlikView Documents

- Q: How much RAM do I need? Is there a simple way to relate RAM requirements to data volume?
- A: No, it depends on the data structure and the type of data. The more rows of data and the wider each record is in terms of number of fields the more RAM is needed. If a field contains many different distinct values more RAM is required than if the number of distinct values is low. Certain charts may demand a large portion of RAM while being calculated.
- Q: What is the theoretical limit of document size in QlikView
- A: In a 32-bit Windows environment QlikView can never be given more than 2 GB of virtual memory (3 GB on Windows Advanced Server) so that is a very definite limit. It is impossible to translate this limit to a certain number of records (see question above). With QlikView 64-bit edition the 2GB limitation disappears. Typically QlikView uses 3-4 times as much memory during script execution as when working in the layout. Running the script in QlikView 64-bit edition but using the document in a 32-bit QlikView can therefore be a workable solution in many cases.
- Q: I have 2 GB of RAM available but my 1 GB document gives me an "Out of memory" error message when opening. What is wrong?

-
- A: Each table in a QlikView document requires allocation of a contiguous chunk of memory. If memory is fragmented, e.g. by loaded dll's, the allocation will fail and a correct error message will be shown. This behavior may vary between different computers.
- Q: How large documents are actually run in QlikView by real users?
- A: When QlikView is run on a standard modern PC with say 512MB of RAM a few million rows of typical transaction data can be handled. In large servers with 64-bit QlikView and several Gigabytes of RAM up to one billion rows (full transaction detail level without aggregation) are handled in customer applications at the time of writing (May 2008).
- Q: Can I run documents created in old QlikView versions in later versions of QlikView?
- A: Yes, you can read all files created in version 7 and later. If you need to convert files from earlier versions of QlikView, you need QlikView 7. Contact your program vendor for a free copy. QlikView 9 shares file format with QlikView 7 and 8.
- Q: Can I run documents created in QlikView 9 in earlier versions of QlikView?
- A: Usually yes! QlikView 9 files can be used directly by QlikView 8 and 7. QlikView 9 can also save files in QlikView 8 and 7 formats, but the formatting and functionality not supported in the earlier version will be lost.

D.3 Scripts and Loading Data

- Q: Is it possible to use more than one Binary statement in a QlikView script?
- A: No unfortunately not. Binary load uses very special logic to read the data already processed once into QlikView in another QlikView document. The "unpacking" used requires a "clean slate" in memory which is the reason why a Binary statement must be the very first statement in the script. Therefore it is also impossible to use Binary

more than once in a script. However you can use QVD files to consolidate data from multiple QlikView documents at the same speed as with Binary.

Q: Is the only requirement for an automatic join that the field names are the same?

A: Yes! Such a join is called an association.

Q: Can I achieve an automatic join between two fields with different field names?

A: Yes, but you must rename one of the fields in the script using an **alias** statement or an **as** clause.

Q: Can I use **where** clauses, **group by** clauses and other SQL-like syntax on text files?

A: QlikView offers you a rich script language for text files including large parts of SQL syntax and a number of additional features.

Q: Can I use **where** clauses, **group by** clauses and other SQL-like syntax on binary QlikView files?

A: No.

Q: What is the difference between an *association* between logical tables, a **load** or **select** statement preceded by a **join** qualifier, and a **join** within a **select** statement?

A: An *association* is a join between logical tables which is evaluated when you click on a field value in your document. QlikView makes this association automatically when two different tables have a field in common. The two latter joins are made during script execution and result in new logical tables. In these cases, QlikView does not load the original tables as logical tables.

Furthermore, a join within a **select** statement will usually only load those records whose key field value is found in both tables (*inner join*), whereas the two former joins also include records whose key field value is found in only one of the tables (*full outer join*).

-
- Q: How do I load data from fix-record text files?
- A: Use the table file wizard in QlikView.
- Q: Can I update a QlikView document by loading only the data that have changed?
- A: Yes, see the chapter QVD files page 497 in Book I for instructions on how to make incremental reloads.
- Q: Can I read tables from web pages into QlikView?
- A: Yes, the table file wizard makes it possible to extract tables when they are coded as tables in HTML. If the HTML page does not contain nicely formatted table tags it is still usually quite possible to extract data using e.g. the **subfield** and **textbetween** script functions.

D.4 QlikView Logic

- Q: Why is it impossible to show frequency in certain list boxes?
- A: The list box in which this problem occurs contains a field which is common to more than one input table. It is therefore impossible for QlikView to know how to calculate frequency and other statistical entities for the field. You can solve the problem by loading the field an extra time from its main table (the one for which you want to show frequency), but under a new name, and show frequency for that field within the list box instead. By using the label feature, the end user does not have to notice the trick.
- Q: Why does my statistics box come up with just a lot of n/a?
- A: The statistics box contains a field which is common to more than one input table. See the answer to the previous question.
- Q: Why doesn't QlikView allow me to use a certain field in an expression in a chart?
- A: The field is common to more than one input table. See the answer to the two previous questions.

-
- Q: How can I display the number of distinct values in a statistics box?
- A: Use the **distinct** clause in your **load** / **select** statements.
- Q: When is the **and** mode option in the **List Box Properties** dialog enabled?
- A: The **and** mode option is only allowed under very strict conditions, imposed by the theory behind the QlikView logic. In order to be used in **and** mode, the field must:
- exist in only one logical table,
 - be the second field of only two fields in that table and
 - contain no duplicate records.
 - it must be proceeded by a distinct qualifier.
- Q: Can I mix **and** and **or** logic for value selections within a list box?
- A: No, the QlikView logic prohibits this.
- Q: Can I have **or** logic between list boxes?
- A: Only indirectly. An advanced alternative is to use semantic links to transfer selections made in a list box to another field. You may then change the selections in the original box and CTRL-click on the semantic link to achieve **or** between the two selections. The final result can be transferred by means of reversing the semantic link.
- Q: Is it possible to link more than one info file to a value, e.g. a picture and a text file?
- A: Yes, but only when duplicates of the field are used. Each field in QlikView can only be linked to one info file.
- Q: Why are for instance '002', '02' and '2' sometimes interpreted as the same value by QlikView?

-
- A: All the values in the example share the same numeric value. As a rule, QlikView will try a numeric interpretation of any data. If a numeric interpretation is possible, it will be used for the association. If you use the interpretation function **text()** on a field in the script, however, the values will be treated strictly as text values. The values in the example above will then be interpreted as three different values.
- Q: What is a "loop" or circular table structure?
- A: When it is possible to follow the field associations in a never ending circle through the table structure, this is called a loop. Another way of describing the phenomenon is that there are two or more different routes through the table structure between two specific fields. Loops should be avoided as far as possible since they may cause ambiguities in the way data is interpreted. In many cases loops are a result of poor database design, but in other cases they may be unavoidable. QlikView will issue a warning if it finds a loop while executing the script and force you to resolve the problem with the help of loosely coupled tables.

D.5 Layout

- Q: Why cannot sheet objects without caption be sized on the top?
- A: When you turn off the caption for a sheet object the upper border will be used for moving the sheet object around. Use the upper corners to size instead.
- Q: Why can't I expand my table to the right when I try to drag the border?
- A: There are two handles on the right border of a table. If you drag the border of a table, you size the outer limits of what can be shown in the table. However, you can never make the outer frame larger than the sum of the widths of all columns in the table. The sizing of the rightmost column is made by putting the cursor just to the left of the outer border. You must however make sure that the outer frame does not fall in the middle of the column. If that is the case, use the scroll bar to position the column's right border at the outer frame. Use the design grid to see the actual outer frame of a sheet object.

Q: Can I change the name of a field in sheet objects in a QlikView document?

A: Yes, you can set a label for each field in each sheet object.

Q: Which sheet objects can be minimized?

A: All types of sheet objects can be minimized, but the option is set to off by default for objects such as buttons, text objects and line/arrow objects.

Q: Can I move minimized objects on the sheet?

A: Yes, they can be moved freely and placed anywhere on the sheet and also sized within certain limits.

D.6 Sharing QlikView Documents with Other People

Q: Can I put my document on a server and share it with other people?

A: Yes, as long as the receivers have a registered QlikView license and access to the server directory.

Q: Can I use QlikView in real client/server mode?

A: Yes, you need a QlikView Server.

Q: Can I put my document as a link on a web page and let others access it over the net?

A: Yes, as long as the receivers have a registered QlikView license and access to the web page.

Q: Can I e-mail my document to other people?

A: Yes, as long as the receivers have a registered QlikView license.

-
- Q: Can I prevent certain people from using my document?
- A: Yes, you can add a "Section Access" to the document to define who may use it.
- Q: Can I prevent other people from seeing/changing the script?
- A: Yes, the "Section Access" can give some people ADMIN access level while keeping other people off the script in USER access level.
- Q: Where do I store information on access rights?
- A: In a text file on a protected unit, as an **inline** statement in the script or as a table in a database.
- Q: Is there an automated way to create personalized copies to a large group of users?
- A: Yes, you need a QlikView Publisher.



E DATA PROTECTION ISSUES

Most of the discussion below assumes that there is an existing database management system (DBMS) and that QlikView is used as an output tool. The arguments, however, still hold true when the raw data are in the form of text files.

E.1 Data Security and Integrity

Data protection usually means several different things. It can mean protecting entered data from being altered or destroyed by mistake, making sure that data is entered in a correct way or preventing the data from being shared by unauthorized people.

The terms integrity and security are often used in this context and although the two concepts seem similar at a first glance, they are in fact quite different. Security refers to the protection of data against unauthorized access, whereas integrity refers to the validity of data, i.e.:

- Security involves assuring that the users are allowed to do what they are trying to do.
- Integrity involves assuring that what the users are trying to do is correct.

We will mostly discuss data security, since the tools for data integrity are provided by the DBMS.

E.2 The Right to Alter Data

The first step in security is to make sure that users cannot erase or change data inadvertently. For multi-user systems this implies using an operating system and a database management system with adequate protection. Examples of such operating systems for PCs are Windows NT or Novell. Examples of such database management systems are ORACLE, SQL Server or Informix.

If data are not protected by the operating system, it will always be possible to delete data by mistake. This is true even if the file is password protected.

The next step is to set up privileges for the authorized users. In a correctly set up system it should be impossible to do anything with the data unless the correct tools are used, i.e. tools that can check whether you are authorized to do what you are trying to do.

For the single user most of the security problems do not exist. It is thus often sufficient to make regular back-ups of the data files.

E.3 The Right to View Data

The final issue in data security concerns the security when handling confidential information. If the security issues above concern the right to alter data, this point rather concerns the right to view data. Most database management systems have means to prevent people from looking at data residing in the database. They cannot, however, prevent people from looking at a copy of the data found in a QlikView file. For this purpose, QlikView has its own means of preventing unauthorized people from viewing data. One must, however, be aware of the fact that the QlikView access restriction tool only concerns the right to see data. QlikView can never prevent users from destroying data with other tools. Only the operating system can.

E.4 Data Integrity

Data integrity implies a structured data flow. Data entry procedures must be set up to make sure that data are entered in a uniform way. A good way to do this is to design forms with the DBMS. Forms also prevent users from entering non valid values, e.g. nonexistent customer numbers, into a database.

When working with single-user databases, one must also be careful not to have more than one person using the database at a time. Multi-user databases can, as the name implies, handle several people editing the database simultaneously.

Another aspect related to this issue is the following: One must always know if a file or a database is the original or a copy. If this is not the case, someone will most certainly start entering data into a copy of the database.

F USING QLIKVIEW DOCUMENTS AS AN INFORMATION SERVER

This appendix is meant to give you an idea of different ways currently used by QlikView users around the world to distribute their documents.

F.1 Information Server

It can sometimes be practical to use an information server. This is a server that acts as an intermediate step between the database and the end users. It could be a stand-alone server, but it could just as well be a set of directories on a common file server containing the original QlikView documents. At least some of the following advice concerning the information server should be taken:

- The operating system should be one in which it is possible to protect files by setting different access rights for different users, e.g. Windows NT.
- All original QlikView documents should be read-only. This means that end users can fetch documents but cannot by mistake destroy them.
- Updates of QlikView files should be run as batch jobs during the night, usually daily, weekly or monthly.
- If text files are extracted from the original database and used as input files for QlikView, they should be stored in a directory that is not shared by the end users. In this way the information cannot be shared by unauthorized people.
- The QlikView internal access restriction should be used on QlikView files.
- If the database containing the access rights is a file stored on the information server, this file must also be protected by the access restriction set by the operating system.

F.2 Extracting Data via ODBC/OLEDB

If there is a stable ODBC or OLEDB driver for the DBMS available, you can usually extract data in a single step process:

- 1 Define an ODBC or OLEDB data source on the information server. This needs to be done only once.

-
- 2 Run QlikView on the information server, using data from the ODBC or OLEDB data source as input; then create QlikView files that end users can use.

F.3 Extracting Data From AS/400 With Router

When the data reside in an AS/400 connected to the PC network via a router, it is usually a good idea to import data into QlikView in a two step process:

- 1 Extract data from the AS/400 to files on an information server. Several SQL based query tools exist, e.g. Netsoft Transfer, IBM Client Access, etc. An appropriate file format to use on the information server is the DIF format. Most AS/400 transfer tools can create DIF files.
- 2 Run QlikView on the information server, using the DIF files as input; then create QlikView files that end users can use.

F.4 Extracting Data From Mainframe, AS/400 etc. Without Router

When the data reside in a mainframe, an S/36 or an AS/400 without a router, it is usually a good idea to import data into QlikView in a three step process:

- 1 Query the database in a standard way on the mainframe, e.g. by the use of a COBOL program, generating files on the mainframe. The file format can e.g. be fix-record or DIF.
- 2 Transfer the files to the information server on the PC network using the normal ways of communication between the two networks.
- 3 Run QlikView on the information server, using the generated files as input; then create QlikView files that end users can use.

F.5 Using QlikView Binary Files

An additional step when creating the QlikView files can be added if end users want more freedom to create their own layout, their own buttons and graphs:

The QlikView file created on the information server can be seen as a file master from which the end users can load data into their local files. This is done by using the **binary** statement, a statement loading the data but not the layout. The created layout will thus remain while the data are updated.

F.6 Aggregation

When the amount of data becomes too large, it is no longer practical to load all of it into QlikView. One can also argue that most end users do not need to see all the details in the data. In both cases, it might be useful to aggregate data over some relevant entities like in the following example.

Example - sales statistics

For many companies the order register contains several millions of order lines each year. It can then be interesting to sum the ordered amount and the number of units instead of showing each individual order line in QlikView. The summation can be made for e.g. each month, each product and each customer. Information about product groups, regions, year, etc. can still be used. With this solution, the amount of data extracted to QlikView will decrease drastically as will the search times.

This summation is called aggregation. Detailed information on the script syntax is found in Book I, under "Load" on page 309 and under "Aggregation Functions" on page 359.

F.7 Sharing QlikView Documents in Workgroups

When building a QlikView document, one is often confronted with the issue of how to structure the information flow. Questions like "Where do I put the original?", "How do I prevent end users from destroying the QlikView document?" and "How do I create automatic updates?" arise.

QlikView documents can easily be distributed and shared between the members of a workgroup, the employees of a company or any other group; the only restriction naturally being that each recipient must have his/her own registered license of QlikView.

F.8 Mailing QlikView Documents

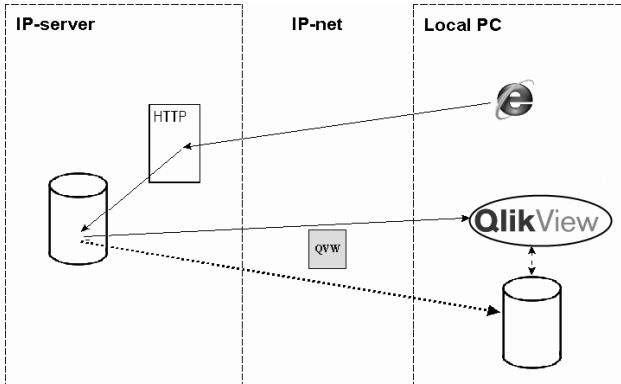
A simple but sometimes efficient approach is, of course, to e-mail QlikView .qvw files. This is quite straightforward and requires no further comment.

F.9 Posting QlikView Documents on a Web Page

A QlikView document can be posted as a link on a web page on an Intranet or on the Internet. The QlikView document file is updated and stored on the server. The recipi-

ents use their web browser to access the page and click on the link. The file can be opened by QlikView directly from the web server or stored on the recipients' local disk.

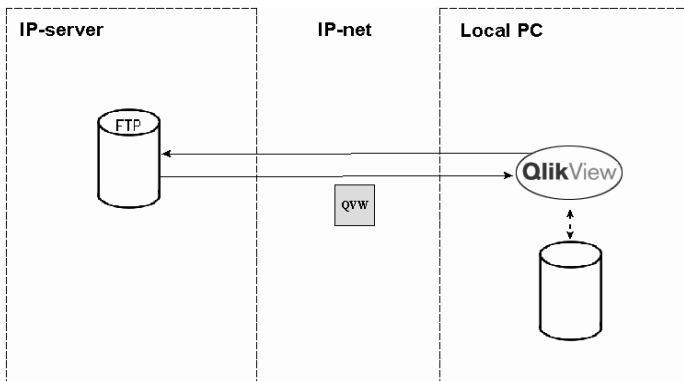
The principle is illustrated in the picture below.



F.10 Posting QlikView Documents on an FTP Server

QlikView documents can be stored in an FTP server and downloaded by recipients over an Intranet or the Internet. The QlikView document file is updated and stored on the server. The recipients use the built-in FTP browser in QlikView to access the document. The file is opened by QlikView directly from the FTP server and can then be stored on the recipients' local disks.

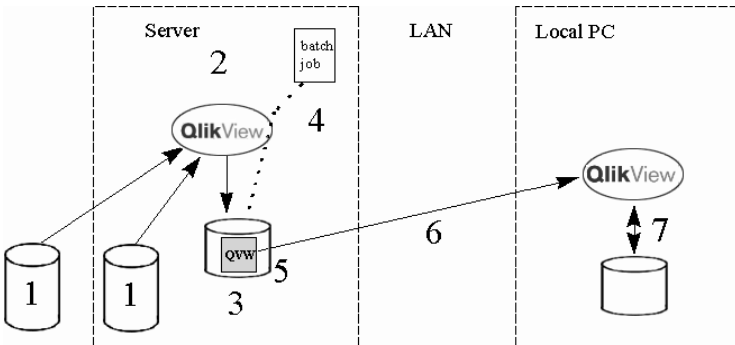
The principle is illustrated in the picture below.



F.11 QlikView Document on Server - Basic Setup

The most obvious and common way to share documents between users is to put the document file on a server in a Local Area Network. A typical setup works in the following way:

- 1 The database(s) containing the original data reside on the server or are accessible from the server.
- 2 The server has its own QlikView license and software for controlling batch jobs.
- 3 The document resides on the server and includes a section access, which gives only the system administrator and the server QlikView license ADMIN access, leaving everyone else on USER level.
- 4 At regular intervals (e.g. every night at 2 PM) a batch job is started, which updates the document by running the script in batch mode and saving the document with the new data.
- 5 The batch job either includes making copies of the document and saving those on different locations accessible to the end users or only stores the original in a read-only directory on the server.
- 6 The end users can access the QlikView document but typically not see or execute the script (because they only have USER level access rights).
- 7 The end user can save the document locally on his/her computer and can make temporary additions to the layout, e.g. make new charts. These will however be lost each time the document on the server is updated, since the end user cannot reload data.



Some advantages of this setup are:

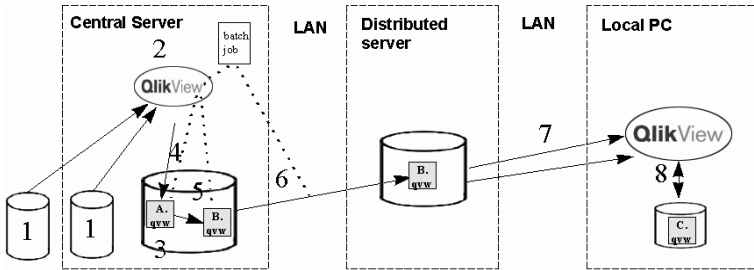
-
- Only one access per day or so is made to the original databases. This reduces both costs for extra simultaneous users and response times in the databases.
 - All end users are forced to use the same document.
 - All end users are guaranteed to find updated versions on the server.

F.12 QlikView Document on Server - Advanced Setup

In some cases there are larger demands on data security. You may also want to allow users to make additions to the document without losing the possibility to reload data. If so, the following setup could be a solution.

- 1 The database(s) containing the original data reside on the server or are accessible from the server.
- 2 The server has its own QlikView license and software for controlling batch jobs.
- 3 A main document (e.g. *a.qvw*) resides on the server. It includes a section access, which prohibits opening unless it is opened by the QlikView license on the server.
- 4 At regular intervals (e.g. every night at 2 PM) a batch job is started, which updates the main document by running the script in batch mode and saving the document with the new data.
- 5 A second document for distribution (e.g. *b.qvw*) resides on the same server. The two documents are identical in everything except the script and access restrictions. The script of the second document contains a **binary** statement loading the internal database from the first document. In the script of the second document there is no reference to the original data sources and it will thus reveal nothing about the structure, passwords etc. of the database. The second document is automatically updated by the batch job as soon as the main document has been updated.
- 6 The batch job finally copies the second document from the restricted central servers to one or more locations on servers and/or workstations on the local network.
- 7 The end user can access the application from the server and possibly save a local copy on his/her workstation.

- 8 The end user can make additions to the layout and/or add own data to his locally saved document. By changing the script slightly, to make the **binary** statement point at the document on the accessible server, data can be reloaded without own additions being lost.



Some advantages of this setup are:

- Only one access per day or so is made to the original databases. This reduces both costs for extra simultaneous users and response times in the databases.
- Super secure. No end user can even see the structure of the original databases.
- All end users are guaranteed to find updated versions on the server.
- End users can make their own additions to the document and still reload data without losing anything.

F.13 Customized QlikView Documents

It is quite possible that one would want to automatically deliver different subsets of one large QlikView document to different persons in an organization. Let us assume that you would like to spread information from the company budget system. You would then probably want to make a separate QlikView document to the manager in charge of each cost center.

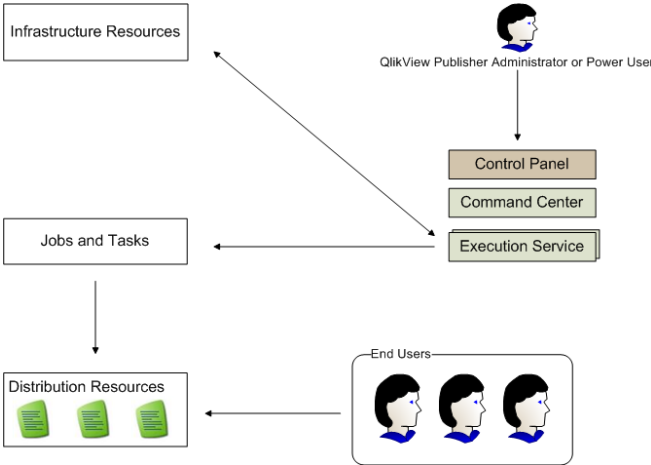
QlikView offers ways of automating this task by means of dividing one large document into several subsets and storing them at predefined locations. For more information about how to achieve this, please contact your local QlikTech organization.

F.14 QlikView Server

QlikView documents can be shared in a real client/server environment either with QlikView as a client or with Java clients running in standard web browsers. Contact your vendor for information on QlikView Server.

F.15 QlikView Publisher

The update and subdivision of QlikView documents can be automated via the QlikView Publisher. This product features a central repository for documents and users combined with an administration panel and a factory module. Using the QlikView AccessPoint you may also create a single point of access for users to all QlikView documents. Contact your vendor for information on QlikView Publisher and QlikView AccessPoint.



G BACKUS-NAUR FORMALISM

The QlikView command line syntax (page 45 in Book I) and script syntax (page 281 in Book I) are described in a notation called Backus-Naur Formalism, or BNF code. Here follows a short description of the BNF code used in this manual:

Symbol interpretation

- | Logical or: the symbol on either side can be used.
- () Brackets defining precedence: used for structuring the BNF syntax.
- [] Square brackets: enclosed items are optional.
- { } Braces: enclosed items may be repeated zero or more times.
- symbol* A non-terminal syntactic category: can be divided further into other symbols, e.g. compounds of the above, other non-terminal symbols, text strings, etc.
- ::= Marks the beginning of a block that defines a symbol.
- load** A terminal symbol consisting of a text string. Should be written as it is into the script.

Non-terminal symbols are indicated by italic type and all terminal symbols are printed in a bold face font. E.g. "(" should be interpreted as a bracket defining precedence, whereas "(" should be interpreted as a character that should be printed in the script.

Example:

The description of the **alias** statement is:

```
alias fieldname as aliasname { , fieldname as aliasname }
```

This should be interpreted as the text string **alias**, followed by an arbitrary field name, followed by the text string **as**, followed by an arbitrary alias name. Any number of additional combinations of *fieldname as aliasname* may be given, separated by commas.

E.g. the following statements are correct:

```
alias a as first;  
alias a as first, b as second;  
alias a as first, b as second, c as third;
```

And the following statements are not correct:

```
alias a as first b as second;  
alias a as first { , b as second};
```




INDEX

Symbols

&	logical operator in chart expr.	296
&	(string concatenation)	296
<\$Endrange	522
<\$Startrange	517

A

Above	388
Accumulation (in chart)	48
Acos	376
AddMonths	418
After	395
Age	433
aggr	special function for advanced aggregations in chart	353
Aggregation functions	in chart expressions	298, 449
Alt	407
And	296
Animation	36
Applycodepage	387
ARGB	443
Arrows	in scatter charts	136
AS/400	extracting data from	516
Asin	376
Atan	376
Atan2	376
Avg	309

B

Backcast	69
Bar Chart	27–86
Bar charts	Bar Settings	55
Bar Offset	41
Basic Aggregation Functions	299
Basic aggregation functions in chart	firstsortedvalue	302
	max	301

min	300
mode	302
only	301
sum	299
Before	393
Below	389
Bit operators in chart	bitand	298
	bitnot	298
	bitor	298
	bitxor	298
Bitand	298
Bitcount	366
Bitnot	298
Bitor	298
Bitxor	298
BlackAndSchole	379
Block chart	177–187
Border	in chart layout	77
Bottom	392
Box Plot Wizard	261

C

Capitalize	385
Caption	in charts	83
	in sheet objects	83
Ceil	364
Chart	13–20
	calculation condition for	22
	drill-down	469
	expressions	293–448
Chart color functions in chart	ARGB	443
	Color	443
	colormix1	444
	colormix2	444
	HSL	445
	RGB	443
	syscolor	447
Chart Expression Syntax	293
Chart Texts	64
Chart texts	64

Chi2-Test	288
Chi2test_chi2	331
Chi2test_df	331
Chi2test_p	330
Chidist	376
Chiinv	377
Chr	383
Class	408
Color	443
in chart	70
Color functions	
in chart expressions	443
Color Map	
current	73
Color map	
document default	73
QlikView default	74
sheet default	73
Colormix Wizard	254
Colormix1	444
Colormix2	444
Column	388
ColumnNo	399
Combin	366
Combo chart	99–111
ComputerName	409
Concat	304
Conditional functions	
in chart expressions	406
Conditional functions in chart	
alt	407
class	408
if	406
match	407
mixmatch	407
pick	407
wildmatch	408
Wildmatch5	408
ConvertToLocalTime	415
Correl	314
Cos	375
Cosh	376
Count	305
Counter Aggregation Functions	305
Counter aggregation functions in chart	
count	305
MissingCount	308
NullCount	307
NumericCount	306
TextCount	307

Creating	
a bar chart	27
a chart	18
a line chart	87
Custom Error Messages	471
Cyclic	
groups	470

D

Data	
security	513–514
Date	436, 440
Date and time functions in chart	
AddMonths	418
age	433
ConvertToLocalTime	415
day	412
DayEnd	426
DayName	427
DayNumberOfQuarter	417
DayNumberOfYear	417
DayStart	426
firstworkdate	434
hour	414
InDay	433
InDayToTime	433
InLunarWeek	432
InLunarWeekToDate	432
InMonth	430
InMonths	430
InMonthsToDate	431
InMonthToDate	430
InQuarter	429
InQuarterToDate	429
InWeek	431
InWeekToDate	431
InYear	428
InYearToDate	429
lastworkdate	434
LocalTime	415
LunarweekEnd	425
LunarWeekName	425
LunarweekStart	424
MakeDate	413
MakeTime	414
MakeWeekDate	413
minute	414
month	412
MonthEnd	421
MonthName	421

MonthsEnd	422	Even	366
MonthsName	423	Examples	
MonthsStart	422	of aggregate qualifiers	449
MonthStart	421	of chart inter-record functions	453
networkdays	434	of ranking functions	450
now	415	Exp	375
QuarterEnd	420	Exponential and logarithmic functions in chart	
QuarterName	420	exp	375
QuarterStart	419	log	375
second	414	log10	375
setdateyear	428	pow	375
setdateyearmonth	428	sqr	375
today	415	sqrt	375
week	412	Exponential functions	
weekday	413	in chart expressions	375
WeekEnd	424	Expressions	
WeekName	424	in chart	293–448
WeekStart	423	Extracting data	
weekyear	413	from AS/400	516
year	412	from mainframe	516
YearEnd	419	F	
YearName	419	Fabs	365
YearStart	418	Fact	365
yeartodate	416	False	383
Date functions		FAQ	503
in chart expressions	412	Fast	24
Day	412	Fdist	378
DayEnd	426	Field groups	469
DayName	427	Fieldindex	388
DayNumberOfQuarter	417	Fieldvalue	387
DayNumberOfYear	417	Fieldvaluecount	387
DayStart	426	Financial	326
Dimensionality	401	Financial aggregation function in chart	
Div	363	irr	326
Document Functions	402	npv	328
Document functions in chart		xirr	327
NoOfReports	403	xnpv	329
ReportComment	402	Financial Aggregation Functions	326
ReportID	403	Financial functions	
ReportName	403	in chart expressions	379
ReportNumber	403	Financial functions in chart	
Drill-down chart	469	BlackAndSchole	379
E		fv	380
e, base of natural logarithms	383	nper	380
Equation		pmt	381
Show	49	pv	382
Error messages		rate	382
custom	471	Findoneof	387

Finv	378
First	396
Firstsortedvalue	302
Firstworkdate	434
Floor	364
Fmod	363
Follows	297
Forecast	69
Formatting functions	
in chart expressions	439
Formatting functions in chart	
date	440
interval	442
money	440
num	439
time	441
timestamp	441
Frac	364
Fractile	311
Frame	
color settings	71
Functions in chart expressions	
aggregation	298, 449
color	443
conditional	406
date and time	412
exponential	375
financial	379
formatting	439
hyperbolic	375
interpretation	434
inter-record	387, 453
logarithmic	375
logical	408
mathematical constants	383
null	409
numeric	363
other	362
range	367
ranking	403, 450
statistical distribution	376
string	383
synthetic dimension	361
system	409
trigonometric	375
Funnel chart	167–176
Fv	380

G

Gauge chart	189–200
-------------------	---------

General numeric functions in chart	
bitcount	366
ceil	364
combin	366
div	363
even	366
fabs	365
fact	365
floor	364
fmod	363
frac	364
mod	363
odd	366
permut	366
round	364
sign	365
GetActiveSheetID	409
GetAlternativeCount	411
GetCurrentField	410
GetCurrentSelections	410
GetExcludedCount	411
GetFieldSelections	410
GetNotSelectedCount	411
GetPossibleCount	411
GetRegistryString	409
GetSelectedCount	411
Grid chart	145–155
Groups	
cyclic	470
hierarchic	469
non-hierarchic	470
Groups of fields	469

H

Help Text	
in charts	86
Hierarchic groups	469
Highlight	122
in combo chart	107
in scatter chart	137
Hour	414
Hrank	405
HSL	445
Hyperbolic functions	
in chart expressions	375

I

If	406
InDay	433

InDayToTime	433
Independent Samples T-Test	291
Index	384
Info	386
InLunarWeek	432
InLunarweekToDate	432
InMonth	430
InMonths	430
InMonthsToDate	431
InMonthToDate	430
inputavg	352
inputsum	351
InQuarter	429
InQuarterToDate	429
Interpretation functions in chart expressions	434
Inter-record functions in chart expressions	387, 453
Inter-record functions in chart above	388
after	395
before	393
below	389
bottom	392
Column	388
ColumnNo	399
dimensionality	401
fieldindex	388
fieldvalue	387
fieldvaluecount	387
first	396
last	397
NoOfColumns	400
NoOfRows	399
RowNo	398
secondarydimensionality	402
top	391
Interval	438, 442
InWeek	431
InWeekToDate	431
InYear	428
InYearToDate	429
Irr	326
IsNull	409
IsNum	408
IsText	409

K

KeepChar	386
Kurtosis	313

L

Last	397
Lastworkdate	434
Layout	77
Left	384
Legend Settings dialog	60
Len	383
Like	296
Line chart	87–98
Line Style in Bar chart	41
Line/Symbol Settings in bar charts	57
in combo charts	107
in line charts	97
Linest_b	317
Linest_df	322
Linest_f	323
Linest_m	316
Linest_r2	318
Linest_seb	320
linest_sem	319
Linest_sey	321
Linest_ssreg	324
Linest_sresid	325
LocalTime	415
Log	375
Log10	375
Logarithmic functions in chart expressions	375
Logical functions in chart expressions	408
Logical functions in chart IsNum	408
IsText	409
Logical operator in charts and	296
not	296
or	296
xor	296
Lower	385
Ltrim	385
LunarweekEnd	425
LunarweekName	425
LunarweekStart	424

M

MakeDate	413
MakeTime	414

MakeWeekDate	413
Match	407
Mathematical constants	
in chart expressions	383
Mathematical constants and parameter-free functions in chart	
e, base of natural logarithms	383
false	383
pi	383
rand	383
true	383
Max	301
MaxString	304
Median	311
Memory	
requirements	488
Mid	384
Min	300
Mini Chart	238
Mini Chart Settings	239
MinString	303
Minute	414
MissingCount	308
Mismatch	407
Mod	363
Mode	302
Money	435, 440
Month	412
MonthEnd	421
MonthName	421
MonthsEnd	422
MonthsName	423
MonthsStart	422
MonthStart	421

N

Nested Aggregations	459
Networkdays	434
Non-hierarchic groups	470
NoOfColumns	400
NoOfReports	403
NoOfRows	399
Normdist	377
Norminv	377
Not	296
Now	415
Nper	380
Npv	328
Null	409
Null functions	

in chart expressions	409
Null functions in chart	
IsNull	409
Null	409
NullCount	307
Num	435, 439
Number interpretation functions in chart	
date	436
interval	438
money	435
num	435
text	438
time	437
timestamp	437
Numbers on Data Points	
in combo charts	109
in line chart	96
Numeric functions	
in chart expressions	363
NumericCount	306

O

Object menu	
for bar chart	27
for block chart	177
for chart	18
for combo chart	99
for gauge chart	189
for grid chart	145
for line chart	87
for pie chart	157
for pivot table	205
for radar chart	113
for scatter chart	127
for straight table	228
Odd	366
Only	301
Operating system	
requirements	487, 488
Operators in chart expressions	
bit	298
logical	296
numeric	295
relational	297
string	296
Or	296
Ord	383
OSuser	409
Other functions	
in chart expressions	362

P	
Paired T-Test	289
Permut	366
pi	383
Pick	407
Pie chart	157–165
Pie popout	41
Pivot table	201–225
using	201
Pmt	381
Pop-up Labels	
for dimension values in bar charts	57
for dimension values in combo charts	107
for dimension values in grid charts	152
for dimension values in line charts	94
for dimension values in pie charts	163
for dimension values in scatter charts	137
Pow	375
Precedes	297
PurgeChar	386
Pv	382
Q	
QlikView Default	
color map	74
QlikView Publisher	522
QlikView Server	521
Qlikviewversion	409
QuarterEnd	420
QuarterName	420
Quick Chart Wizard	265
QuarterStart	419
QVuser	409
R	
Radar Chart	113–125
Rand	383
Range functions	
in chart expressions	367
Range functions in chart	
rangeavg	367
rangecorrel	373
rangecount	367
rangefractile	369
rangeirr	373
rangekurtosis	369
rangemax	368
rangemaxstring	372
rangemin	368
rangeminstring	372
rangemissingcount	371
rangemode	371
rangenvp	374
rangennullcount	370
rangenumERICcount	370
rangeonly	372
rangeskew	369
rangestdev	368
rangesum	367
rangertextcount	370
rangexirr	374
rangexnpv	374
Rangeavg	367
Rangecorrel	373
Rangecount	367
Rangefractile	369
Rangeirr	373
Rangekurtosis	369
Rangemax	368
Rangemaxstring	372
Rangemin	368
Rangeminstring	372
Rangemissingcount	371
Rangemode	371
Rangenvp	374
Rangennullcount	370
RangenumERICcount	370
Rangeonly	372
Rangeskew	369
Rangestdev	368
Rangesum	367
Rangertextcount	370
Rangexirr	374
Rangexnpv	374
Rank	403
Ranking functions	
in chart expressions	403, 450
Ranking functions in chart	
hrank	405
rank	403
vrack	406
Rate	382
Reference Lines	62
Reference Lines dialog	139
Relational operators in chart	

follows	297
precedes	297
ReloadTime	409
Repeat	385
Replace	386
ReportComment	402
ReportID	403
ReportName	403
ReportNumber	403
RGB	443
Right	384
Round	364
RowNo	398
Rtrim	385

S

Scatter chart	127–144
Second	414
Secondarydimensionality	402
Security	
data protection issues	513–514
Set Expressions	353
Setdateyear	428
Setdateyearmonth	428
Show R2	50
Show value	
in bar chart	42
Sign	365
Sin	376
Sinh	376
Skew	312
Special Function for Advanced Aggregations	353
SqlValue	410
Sqr	375
Sqrt	375
Statistical Aggregation Functions	309
Statistical aggregation functions in chart	
avg	309
correl	314
fractile	311
kurtosis	313
linest_b	317
linest_df	322
linest_f	323
linest_m	316
linest_r2	318
linest_seb	320
linest_sem	319
linest_sey	321
linest_ssreg	324
linest_ssresid	325
median	311
skew	312
stdev	310
sterr	314
steyx	315
Statistical distribution functions	
in chart expressions	376
Statistical distribution functions in chart	
chidist	376
chiinv	377
fdist	378
finv	378
normdist	377
norminv	377
tdist	378
tinv	378
Statistical Test Functions	330
Statistical test functions in charts	
chi2test_chi2	331
chi2test_df	331
chi2test_p	330
TTest_conf	334
TTest_df	332
TTest_dif	333
TTest_lower	335
TTest_sig	333
TTest_sterr	334
TTest_t	332
TTest_upper	335
TTest1_conf	341
TTest1_df	340
TTest1_dif	341
TTest1_lower	342
TTest1_sig	340
TTest1_sterr	341
TTest1_t	339
TTest1_upper	342
TTest1w_conf	345
TTest1w_df	343
TTest1w_dif	344
TTest1w_lower	345
TTest1w_sig	344
TTest1w_sterr	344
TTest1w_t	343
TTest1w_upper	346
TTestw_conf	338
TTestw_df	336
TTestw_dif	337

TTestw_lower	338	trim	385
TTestw_sig	337	upper	385
TTestw_sterr	337	String operators in charts	
TTestw_t	336	like	296
TTestw_upper	339	Subfield	386
ZTest_conf	348	Substringcount	387
ZTest_dif	347	Sum	299
ZTest_sig	347	Synthetic dimension functions	
ZTest_sterr	347	in chart expressions	361
ZTest_z	346	Synthetic dimension functions in chart	
ZTestw_conf	350	valuelist	361
ZTestw_dif	349	valueloop	362
ZTestw_sig	349	Syscolor	447
ZTestw_sterr	350	System	
ZTestw_z	348	requirements	487
Statistics Chart Wizard	287	System functions	
Stdev	310	in chart expressions	409
Sterr	314	System functions in chart	
Steyx	315	ComputerName	409
Straight table	227–247	GetActiveSheetID	409
using	227	GetAlternativeCount	411
String Aggregation Functions	303	GetCurrentField	410
String aggregation functions in chart		GetCurrentSelections	410
concat	304	GetExcludedCount	411
MaxString	304	GetFieldSelections	410
MinString	303	GetNotSelectedCount	411
String functions		GetPossibleCount	411
in chart expressions	383	GetRegistryString	409
String functions in chart		GetSelectedCount	411
applycodepage	387	OSuser	409
capitalize	385	qlikviewversion	409
chr	383	QVuser	409
findoneof	387	ReloadTime	409
index	384	SqlValue	410
info	386		
KeepChar	386	T	
left	384	Tan	376
len	383	Tanh	376
lower	385	Tdist	378
ltrim	385	Text	438
mid	384	Textbetween	386
ord	383	TextCount	307
PurgeChar	386	The Advanced Color Map dialog	73
repeat	385	Time	437, 441
replace	386	Time Chart Wizard	277–286
right	384	Time functions	
rtrim	385	in chart expressions	412
subfield	386	Timestamp	437, 441
substringcount	387	Tinv	378
textbetween	386		

Title	25
Today	415
Top	391
Trend lines	49
Trigonometric and hyperbolic functions in chart	
acos	376
asin	376
atan	376
atan2	376
cos	375
cosh	376
sin	376
sinh	376
tan	376
tanh	376
Trigonometric functions in chart expressions	375
Trim	385
Troubleshooting (FAQ)	503
True	383
Ttest_conf	334
Ttest_df	332
Ttest_dif	333
Ttest_lower	335
Ttest_sig	333
Ttest_sterr	334
Ttest_t	332
Ttest_upper	335
Ttest1_conf	341
Ttest1_df	340
Ttest1_dif	341
Ttest1_lower	342
Ttest1_sig	340
Ttest1_sterr	341
Ttest1_t	339
Ttest1_upper	342
Ttest1w_conf	345
Ttest1w_df	343
Ttest1w_dif	344
Ttest1w_lower	345
Ttest1w_sig	344
Ttest1w_sterr	344
Ttest1w_t	343
Ttest1w_upper	346
Ttestw_conf	338
Ttestw_df	336
Ttestw_dif	337
Ttestw_lower	338
Ttestw_sig	337

Ttestw_sterr	337
Ttestw_t	336
Ttestw_upper	339

U

Upper	385
Use borders	77

V

Valuelist	361
Valueloop	362
Values on Data Points in bar charts	59
Virtual memory	488
Vrank	406

W

Waterfall charts	41
Week	412
Weekday	413
WeekEnd	424
WeekName	424
WeekStart	423
Weekyear	413
Wildmatch	408
Wildmatch5	408

X

Xirr	327
Xnpv	329
Xor	296

Y

Year	412
YearEnd	419
YearName	419
YearStart	418
Yeartodate	416

Z

Zero on Bars	57
Ztest_conf	348
Ztest_dif	347
Ztest_sig	347
Ztest_sterr	347
Ztest_z	346
Ztestw_conf	350

Ztestw_dif	349
Ztestw_sig	349
Ztestw_sterr	350
Ztestw_z	348