

# QlikView

## Reference Manual

*Version 11.20 SR6 for Microsoft Windows®*

*Lund, Sweden, 2014*

*Authored by QlikTech International AB*

Copyright © 1994-2014 Qlik®Tech International AB. All rights reserved.

Qlik™, QlikView™, QlikTech™, and the QlikTech logo are trademarks of QlikTech International AB which have been registered in multiple countries. Other trademarks referenced herein are the trademarks of their respective owners.

# Contents

<b>Part 1 Introduction</b> .....	<b>17</b>
<b>1 Before You Begin</b> .....	<b>19</b>
1.1 Conventions.....	19
1.2 QlikTech Support Services.....	20
<b>2 What's New in QlikView 11.20?</b> .....	<b>21</b>
2.1 General Features.....	21
<b>3 Compatibility Issues for Earlier Versions of QlikView</b> .....	<b>23</b>
3.1 File Format Compatibility.....	23
3.2 Mixed Client/Server Environments.....	23
<b>4 QlikView Personal Edition</b> .....	<b>25</b>
<b>Part 2 Installation</b> .....	<b>27</b>
<b>5 Installing QlikView</b> .....	<b>28</b>
5.1 QlikView Installation Packages.....	28
5.2 Downloading and Starting the QlikView Desktop Installation Program.....	29
5.3 When the Installation Program Is Running.....	29
5.4 Serialization.....	29
5.5 Network Distribution of QlikView.....	30
5.6 License Enabler File Editor.....	30
5.7 Updating QlikView.....	31
<b>6 OLE DB and ODBC</b> .....	<b>33</b>
6.1 OLE DB.....	33
6.2 When Is the ODBC Interface Needed?.....	33
6.3 Creating ODBC Data Sources.....	34
<b>Part 3 Navigation and Basics</b> .....	<b>35</b>
<b>7 Starting QlikView</b> .....	<b>36</b>
7.1 Command Line Syntax.....	36
7.2 The Start Page.....	36
7.3 Connect to Server.....	38
7.4 Batch Execution.....	39
<b>8 Getting Started Wizard</b> .....	<b>43</b>
<b>9 QlikView File</b> .....	<b>45</b>
9.1 QlikView Project Files.....	45
9.2 Source Control.....	46
<b>10 Menu Commands</b> .....	<b>47</b>
10.1 File Menu.....	47
10.2 Edit Menu.....	51
10.3 View Menu.....	52
10.4 Selections Menu.....	53
10.5 Layout Menu.....	53
10.6 Settings Menu.....	54
10.7 Bookmarks Menu.....	55
10.8 Reports Menu.....	55
10.9 Tools Menu.....	55

10.10 Object Menu .....	56
10.11 Window Menu .....	56
10.12 Help Menu .....	57
<b>11 Toolbars and Status Bar .....</b>	<b>59</b>
11.1 Toolbars .....	59
11.2 Standard Toolbar .....	59
11.3 Navigation Toolbar .....	61
11.4 Design Toolbar .....	61
11.5 Sheet Toolbar .....	64
11.6 Bookmarks Toolbar .....	64
11.7 Customize (Toolbars).....	64
11.8 Status Bar .....	67
<b>12 User Preferences .....</b>	<b>69</b>
12.1 User Preferences: General .....	69
12.2 User Preferences: Save .....	73
12.3 User Preferences: Editor .....	74
12.4 User Preferences: Design .....	77
12.5 User Preferences: Objects .....	78
12.6 Font .....	80
12.7 User Preferences: Export .....	81
12.8 User Preferences: Printing .....	83
12.9 User Preferences: Mail .....	84
12.10 User Preferences: Locations .....	85
12.11 User Preferences: Security .....	87
12.12 User Preferences: License .....	88
<b>13 Exporting and Printing .....</b>	<b>91</b>
13.1 Print: General .....	91
13.2 Print: Layout .....	92
13.3 Print: Header/Footer .....	93
13.4 Date & Time .....	95
13.5 Print Sheet .....	95
13.6 Print Preview .....	96
13.7 Copy Mode .....	96
13.8 Export/Export Contents.....	97
<b>14 Logic and Selections .....</b>	<b>99</b>
14.1 Single Field Value Selection .....	99
14.2 Color Codes .....	99
14.3 Selection Styles .....	99
14.4 Indicators .....	101
14.5 Multiple Selection Within a Field .....	101
14.6 Moving Selections .....	102
14.7 Locking Selected Field Values .....	103
14.8 Current Selections .....	103
14.9 Selections in Other Objects .....	103
14.10 Search .....	105
14.11 Advanced Search Dialog .....	108
14.12 AND-Mode in List Boxes .....	110
14.13 Circular References .....	111

14.14 Loosely Coupled Tables.....	113
14.15 Alternate States.....	113
<b>15 Bookmarks.....</b>	<b>117</b>
15.1 Bookmark Types.....	117
15.2 Add Bookmark Dialog.....	117
15.3 Bookmarks Dialog.....	119
15.4 Export Bookmark(s).....	121
15.5 Import Bookmarks.....	122
15.6 Alternate States and Bookmarks.....	123
<b>16 Reports.....</b>	<b>125</b>
16.1 Reports Menu.....	125
16.2 Report Editor.....	126
<b>17 Alerts.....</b>	<b>141</b>
17.1 Creating Alerts.....	141
17.2 Using Alerts.....	141
17.3 Alerts.....	142
17.4 The Alerts Dialog.....	143
17.5 Alert Wizard.....	146
<b>18 Variable Overview.....</b>	<b>153</b>
<b>19 Expression Overview.....</b>	<b>155</b>
<b>20 Internal Files.....</b>	<b>157</b>
20.1 File References in the QlikView Layout.....	157
20.2 Internal Files in QlikView Documents.....	157
20.3 Internal Files in the QlikView Program File.....	157
20.4 Sounds.....	158
<b>Part 4 Script.....</b>	<b>159</b>
<b>21 Introduction to Loading Data.....</b>	<b>160</b>
21.1 Loading Data from Files.....	160
21.2 Loading Data from Databases.....	160
21.3 Direct Discovery.....	161
21.4 Differences between Direct Discovery and In-Memory Data.....	165
21.5 Direct Discovery with QlikView Server and Publisher.....	169
<b>22 Variables and Fields.....</b>	<b>171</b>
22.1 Fields.....	171
22.2 Input Fields.....	172
22.3 Variable.....	173
22.4 Other Entities in QlikView.....	173
<b>23 Edit Script Dialog.....</b>	<b>175</b>
23.1 Set Statement Wizard.....	181
23.2 Find/Replace (Script).....	182
23.3 Hidden Script.....	183
23.4 Table Viewer.....	183
23.5 Data Link Properties.....	186
23.6 Connect to Data Source.....	188
23.7 Create Select Statement.....	189
23.8 Open Local Files.....	192
23.9 Open Internet Files or Open QlikView Document.....	193

23.10	Inline Data Wizard	194
23.11	Access Restriction Table Wizard	195
23.12	File Wizard	196
<b>24</b>	<b>Script Syntax</b>	<b>219</b>
24.1	Script Statements and Keywords	219
24.2	Table Names	281
24.3	Script Variables	282
24.4	Dollar-Sign Expansions	290
24.5	Quotation Marks in Scripting	292
<b>25</b>	<b>Script Expressions</b>	<b>295</b>
25.1	Operators	295
25.2	Aggregation Functions	298
25.3	Other Functions	315
<b>26</b>	<b>Data Structures</b>	<b>379</b>
26.1	Data Loading Statements	379
26.2	Execution of the Script	379
26.3	System Fields	379
26.4	Logical Tables	381
26.5	Associations between Logical Tables	381
26.6	Renaming Fields	384
26.7	Concatenating Several Tables into One	385
26.8	Join and Keep	386
<b>27</b>	<b>Evaluating the Loaded Data</b>	<b>391</b>
27.1	Generic Databases	391
27.2	Cross Tables	392
27.3	Matching Intervals to Discrete Data	394
27.4	Using the Extended IntervalMatch Syntax to Resolve Slowly Changing Dimension Problems	395
27.5	Hierarchies	397
27.6	Semantic Links	398
27.7	Linking Information to Field Values	401
27.8	Data Cleansing	402
27.9	Wild Cards in the Data	404
27.10	NULL Value Handling in QlikView	405
27.11	Character Set	407
<b>28</b>	<b>QVD Files</b>	<b>409</b>
28.1	Purpose of QVD Files	409
28.2	Create QVD Files	409
28.3	Reading Data from QVD Files	410
28.4	Using QVD Files for Incremental Load	410
<b>29</b>	<b>Security</b>	<b>417</b>
29.1	Authentication and Authorization	417
29.2	Security Using the QlikView Publisher	417
29.3	Security Using the Section Access in the QlikView Script	417
29.4	Sections in the Script	417
29.5	Access Levels in Section Access	418
29.6	Section Access System Fields	418
29.7	Restrictions on QlikView Functionality	420
29.8	Dynamic Data Reduction	420

---

29.9 Inherited Access Restrictions .....	421
29.10 Encryption .....	421
<b>Part 5 Sheet and Sheet Objects .....</b>	<b>423</b>
<b>30 Document Properties .....</b>	<b>424</b>
30.1 Document Properties: General .....	424
30.2 Document Properties: Opening .....	430
30.3 Document Properties: Sheets .....	432
30.4 Document Properties: Server .....	434
30.5 Input Box Properties: Constraints and Document Properties: Variables .....	436
30.6 Document Properties: Security .....	438
30.7 Document Properties: Triggers .....	440
30.8 Document Properties: Groups .....	442
30.9 Document Properties: Tables .....	444
30.10 Document Properties: Sort .....	447
30.11 Document Properties: Presentation .....	448
30.12 Document Properties: Number .....	449
30.13 Document Properties: Scrambling .....	451
30.14 Document Properties: Extensions .....	452
30.15 Font .....	453
30.16 Layout .....	454
30.17 Caption .....	459
<b>31 The Sheet .....</b>	<b>461</b>
31.1 Creation .....	461
31.2 Navigation .....	461
31.3 Sheet: Object Menu .....	462
31.4 Sheet Properties .....	462
31.5 Sheet Properties: General .....	463
31.6 Select Fields/Sheet Properties: Fields .....	465
31.7 Sheet Properties: Objects .....	466
31.8 Sheet Properties: Security .....	467
31.9 Sheet Properties: Triggers .....	468
31.10 Tabrow Properties .....	469
<b>32 Sheet Objects .....</b>	<b>471</b>
32.1 Local Objects and Server Objects .....	471
32.2 Object Menu .....	471
32.3 Notes and Comments .....	472
32.4 Drag and Drop to Microsoft Office .....	472
<b>33 List Box .....</b>	<b>473</b>
33.1 List Box: Object Menu .....	473
33.2 List Box Properties: General .....	476
33.3 List Box Properties: Expressions .....	479
33.4 List Box Properties: Sort .....	486
33.5 List Box Properties: Presentation .....	487
33.6 Properties: Number .....	490
33.7 Font .....	491
33.8 Layout .....	492
33.9 Caption .....	496
33.10 List Box Tree View .....	498

---

33.11 How To Create List Boxes and Table Boxes with Direct Discovery.....	499
<b>34 Statistics Box.....</b>	<b>501</b>
34.1 Statistics Box: Object Menu.....	501
34.2 Statistics Box Properties: General.....	503
34.3 Statistics Box Properties: Number.....	505
34.4 Font.....	506
34.5 Layout.....	506
34.6 Caption.....	506
<b>35 Multi Box.....</b>	<b>507</b>
35.1 Multi Box: Object Menu.....	507
35.2 Multi Box Properties: General.....	510
35.3 Multi Box Properties: Sort.....	512
35.4 Multi Box Properties: Presentation.....	513
35.5 Multi Box Properties: Number.....	517
35.6 Font.....	518
35.7 Layout.....	518
35.8 Caption.....	519
<b>36 Table Box.....</b>	<b>521</b>
36.1 Using the Table Box.....	521
36.2 Table Box: Object Menu.....	521
36.3 Table Box Properties: General.....	525
36.4 Table Box Properties: Sort.....	527
36.5 Table Box Properties: Presentation.....	528
36.6 Chart Properties: Style.....	530
36.7 Table Box Properties: Number.....	531
36.8 Font.....	532
36.9 Layout.....	532
36.10 Caption.....	532
<b>37 Current Selections Box.....</b>	<b>533</b>
37.1 Current Selections Box: Object menu.....	533
37.2 Current Selections Box Properties: General.....	535
37.3 Font.....	537
37.4 Layout.....	537
37.5 Caption.....	537
<b>38 Input Box.....</b>	<b>539</b>
38.1 Using the Input Box.....	539
38.2 Input Box: Object Menu.....	539
38.3 Input Box Properties: General.....	541
38.4 Input Box Properties: Presentation.....	543
38.5 Input Box Properties: Constraints and Document Properties: Variables.....	544
38.6 Input Box Properties: Number.....	546
38.7 Font.....	547
38.8 Layout.....	547
38.9 Caption.....	547
<b>39 Button.....</b>	<b>549</b>
39.1 Button: Object Menu.....	549
39.2 Button Properties: General.....	550
39.3 Actions.....	552



---

39.4	Font .....	559
39.5	Layout .....	559
39.6	Caption .....	559
<b>40</b>	<b>Text Object .....</b>	<b>561</b>
40.1	Text Object: Object Menu .....	561
40.2	Text Object Properties: General .....	562
40.3	Actions .....	565
40.4	Font .....	566
40.5	Layout .....	566
40.6	Caption .....	566
<b>41</b>	<b>Line/Arrow Object .....</b>	<b>567</b>
41.1	Line/Arrow Object: Object Menu .....	567
41.2	Line/Arrow Object Properties: General .....	568
41.3	Actions .....	569
41.4	Layout .....	569
41.5	Caption .....	569
<b>42</b>	<b>Slider/Calendar Object .....</b>	<b>571</b>
42.1	Using a Slider/Calendar Object .....	571
42.2	Slider/Calendar Object: Object Menu .....	571
42.3	Slider/Calendar Object Properties: General .....	573
42.4	Slider/Calendar Object Properties: Presentation (Slider Mode) .....	575
42.5	Slider/Calendar Object Properties: Presentation (Calendar Mode) .....	576
42.6	Sort .....	577
42.7	Number .....	577
42.8	Font .....	577
42.9	Layout .....	577
42.10	Caption .....	577
<b>43</b>	<b>Bookmark Object .....</b>	<b>579</b>
43.1	Bookmark Objects in Alternate States .....	579
43.2	Bookmark Object: Object Menu .....	579
43.3	Bookmark Object Properties: General .....	580
43.4	Font .....	582
43.5	Layout .....	582
43.6	Caption .....	582
<b>44</b>	<b>Search Object .....</b>	<b>583</b>
44.1	Search Object: Object Menu .....	583
44.2	Search Object Properties: General .....	584
44.3	Search Object Properties: Presentation .....	586
44.4	Search Object Properties: Sort .....	587
44.5	Font .....	587
44.6	Layout .....	587
44.7	Caption .....	588
<b>45</b>	<b>Container .....</b>	<b>589</b>
45.1	Container: Object Menu .....	589
45.2	Container Properties: General .....	590
45.3	Container Properties: Presentation .....	592
45.4	Font .....	593
45.5	Layout .....	593

45.6 Caption .....	593
<b>46 Custom Object .....</b>	<b>595</b>
46.1 Custom Object: Object Menu .....	595
46.2 Custom Object Properties: General .....	596
46.3 Layout .....	597
46.4 Caption .....	597
<b>47 Server Objects Pane .....</b>	<b>599</b>
47.1 Configuring the Pane .....	599
47.2 Shared Objects .....	599
47.3 Adding and Sharing Objects .....	599
<b>48 Layout Themes .....</b>	<b>601</b>
48.1 About QlikView Layout Themes .....	601
48.2 Applying Themes in the Layout .....	601
48.3 Theme Maker Wizard .....	603
<b>Part 6 Charts .....</b>	<b>607</b>
<b>49 Introduction .....</b>	<b>608</b>
49.1 New Chart .....	610
49.2 Selections in Charts and Tables .....	611
49.3 Fast Chart Type Change .....	612
49.4 Sizing and Moving Chart Components .....	612
49.5 Chart Properties .....	612
49.6 Chart Properties: General .....	613
<b>50 Bar Chart .....</b>	<b>617</b>
50.1 New Chart .....	617
50.2 Bar Chart: Object Menu .....	617
50.3 General .....	619
50.4 Chart Properties: Dimensions .....	620
50.5 Chart Properties: Dimension Limits .....	625
50.6 Chart Properties: Expressions .....	629
50.7 Chart Properties: Sort .....	639
50.8 Chart Properties: Style .....	640
50.9 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart) .....	643
50.10 Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart) .....	649
50.11 Chart Properties: Colors .....	652
50.12 Chart Properties: Number .....	655
50.13 Font .....	656
50.14 Layout .....	656
50.15 Caption .....	656
<b>51 Line Chart .....</b>	<b>657</b>
51.1 New Chart .....	657
51.2 Line Chart: Object Menu .....	658
51.3 General .....	660
51.4 Dimensions .....	660
51.5 Dimension Limits .....	660
51.6 Expressions .....	660
51.7 Sort .....	660
51.8 Style .....	660
51.9 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart) .....	661

---

51.10 Axes.....	663
51.11 Colors.....	663
51.12 Number.....	664
51.13 Font.....	664
51.14 Layout.....	664
51.15 Caption.....	664
<b>52 Combo Chart.....</b>	<b>665</b>
52.1 Combo Chart: Object Menu.....	665
52.2 General.....	667
52.3 Dimensions.....	667
52.4 Dimension Limits.....	667
52.5 Expressions.....	668
52.6 Sort.....	668
52.7 Style.....	668
52.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart).....	668
52.9 Axes.....	671
52.10 Colors.....	671
52.11 Number.....	671
52.12 Font.....	671
52.13 Layout.....	671
52.14 Caption.....	671
<b>53 Radar Chart.....</b>	<b>673</b>
53.1 Radar Chart: Object Menu.....	673
53.2 General.....	675
53.3 Dimensions.....	675
53.4 Dimension Limits.....	675
53.5 Expressions.....	676
53.6 Sort.....	676
53.7 Style.....	676
53.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart).....	676
53.9 Axes.....	679
53.10 Colors.....	679
53.11 Number.....	679
53.12 Font.....	679
53.13 Layout.....	679
53.14 Caption.....	679
<b>54 Scatter Chart.....</b>	<b>681</b>
54.1 Scatter Chart: Object Menu.....	681
54.2 General.....	683
54.3 Dimensions.....	683
54.4 Dimension Limits.....	683
54.5 Chart Properties: Expressions (Scatter Chart).....	684
54.6 Sort.....	684
54.7 Style.....	685
54.8 Chart Properties: Presentation (Scatter- Grid Chart).....	685
54.9 Chart Properties: Axes (Scatter- Grid Chart).....	689
54.10 Colors.....	691
54.11 Number.....	691

---

54.12	Font .....	691
54.13	Layout .....	691
54.14	Caption .....	691
<b>55</b>	<b>Grid Chart .....</b>	<b>693</b>
55.1	Grid Chart: Object Menu .....	693
55.2	Dimensions .....	695
55.3	Dimension Limits .....	695
55.4	Expressions .....	695
55.5	Sort .....	696
55.6	Style .....	696
55.7	Chart Properties: Presentation (Scatter- Grid Chart) .....	696
55.8	Axes .....	700
55.9	Colors .....	700
55.10	Number .....	700
55.11	Font .....	700
55.12	Layout .....	700
55.13	Caption .....	700
<b>56</b>	<b>Pie Chart .....</b>	<b>701</b>
56.1	Pie Chart: Object Menu .....	701
56.2	Dimensions .....	703
56.3	Dimension Limits .....	703
56.4	Expressions .....	703
56.5	Sort .....	704
56.6	Style .....	704
56.7	Chart Properties: Presentation (Pie Chart) .....	704
56.8	Colors .....	705
56.9	Number .....	705
56.10	Font .....	705
56.11	Layout .....	706
56.12	Caption .....	706
<b>57</b>	<b>Funnel Chart .....</b>	<b>707</b>
57.1	Funnel Chart: Object Menu .....	707
57.2	General .....	709
57.3	Dimensions .....	709
57.4	Dimension Limits .....	709
57.5	Expressions .....	710
57.6	Sort .....	710
57.7	Style .....	710
57.8	Chart Properties: Presentation (Funnel Chart) .....	710
57.9	Colors .....	711
57.10	Number .....	711
57.11	Font .....	712
57.12	Layout .....	712
57.13	Caption .....	712
<b>58</b>	<b>Block Chart .....</b>	<b>713</b>
58.1	Block Chart: Object Menu .....	713
58.2	General .....	715
58.3	Dimensions .....	715

---

58.4 Dimension Limits.....	715
58.5 Expressions.....	716
58.6 Style.....	716
58.7 Chart Properties: Presentation (Block Chart).....	716
58.8 Colors.....	718
58.9 Number.....	718
58.10 Font.....	718
58.11 Layout.....	718
58.12 Caption.....	718
<b>59 Gauge Chart.....</b>	<b>719</b>
59.1 Gauge Chart: Object Menu.....	719
59.2 General.....	721
59.3 Dimensions.....	721
59.4 Expressions.....	721
59.5 Sort.....	722
59.6 Style.....	722
59.7 Chart Properties: Presentation (Gauge Chart).....	722
59.8 Actions.....	725
59.9 Colors.....	725
59.10 Number.....	725
59.11 Font.....	725
59.12 Layout.....	726
59.13 Caption.....	726
<b>60 Mekko Chart.....</b>	<b>727</b>
60.1 Mekko Chart: Object Menu.....	727
60.2 General.....	729
60.3 Dimensions.....	729
60.4 Dimension Limits.....	729
60.5 Expressions.....	729
60.6 Sort.....	729
60.7 Style.....	729
60.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart).....	730
60.9 Axes.....	732
60.10 Colors.....	732
60.11 Number.....	733
60.12 Font.....	733
60.13 Layout.....	733
60.14 Caption.....	733
<b>61 Pivot Table.....</b>	<b>735</b>
61.1 Using the Pivot Table.....	735
61.2 Pivot Table: Object Menu.....	738
61.3 General.....	741
61.4 Dimensions.....	741
61.5 Expressions.....	741
61.6 Sort.....	741
61.7 Chart Properties: Presentation (Pivot Table).....	742
61.8 Chart Properties: Visual Cues.....	744
61.9 Chart Properties: Style.....	745

61.10	Number	746
61.11	Font	746
61.12	Layout	746
61.13	Caption	746
<b>62</b>	<b>Straight Table</b>	<b>747</b>
62.1	Using the Straight Table	747
62.2	Straight Table: Object Menu	747
62.3	General	750
62.4	Dimensions	750
62.5	Dimension Limits	750
62.6	Expressions	750
62.7	Chart Properties: Sort (Straight Table)	751
62.8	Chart Properties: Presentation (Straight Table)	752
62.9	Chart Properties: Visual Cues	754
62.10	Chart Properties: Style	755
62.11	Number	756
62.12	Font	756
62.13	Layout	756
62.14	Caption	756
<b>63</b>	<b>Edit Expression Dialog</b>	<b>757</b>
63.1	File Menu	759
63.2	Edit Menu	760
63.3	Settings Menu	760
63.4	Colormix Wizard	760
<b>64</b>	<b>Boxplot Wizard</b>	<b>765</b>
64.1	Boxplot Wizard Define Data	765
64.2	Boxplot Wizard Presentation	766
<b>65</b>	<b>Quick Chart Wizard</b>	<b>767</b>
65.1	Starting the Quick Chart Wizard	767
<b>66</b>	<b>Time Chart Wizard</b>	<b>775</b>
66.1	Starting the Time Chart Wizard	775
<b>67</b>	<b>Statistics Chart Wizard</b>	<b>781</b>
67.1	Starting the Statistics Chart Wizard	781
67.2	Chi2-Test	781
67.3	Paired T-Test	782
67.4	Independent Samples T-Test	783
<b>67</b>	<b>How To Create Charts with Direct Discovery</b>	<b>784</b>
<b>68</b>	<b>Chart Expressions</b>	<b>785</b>
68.1	Alternate States and Chart Expressions	786
68.2	Variable Text Expansion	787
68.3	Operators	787
68.4	Chart Aggregation functions	790
68.5	Synthetic Dimension Functions	825
68.6	Other Functions	825
<b>69</b>	<b>Examples</b>	<b>853</b>
69.1	Examples of Aggregate Qualifiers	853
69.2	Examples of Concat Functions	854
69.3	Examples of Alternate States in Chart Expressions	855

69.4	Examples of Chart Ranking Functions.....	857
69.5	Examples of Chart Inter Record Functions.....	859
<b>70</b>	<b>Nested Aggregations and Related Issues.....</b>	<b>865</b>
<b>71</b>	<b>Calculated Formulas.....</b>	<b>871</b>
71.1	Entering a Calculated Formula.....	871
71.2	Error Messages.....	871
71.3	Calculated Formula.....	871
<b>72</b>	<b>Field Groups.....</b>	<b>873</b>
72.1	Hierarchic Groups (Drill-Down).....	873
72.2	Non-Hierarchic Groups (Cyclic).....	874
<b>73</b>	<b>Custom Error Messages.....</b>	<b>875</b>
<b>Part 7</b>	<b>Number Formats.....</b>	<b>877</b>
<b>74</b>	<b>Data Types in QlikView.....</b>	<b>878</b>
74.1	Data Representation inside QlikView.....	878
74.2	Number Interpretation.....	878
74.3	Dates and Times.....	881
<b>75</b>	<b>Format Codes for Interpretation and Formatting Functions.....</b>	<b>885</b>
75.1	Numbers.....	885
75.2	Dates.....	885
75.3	Times.....	886
75.4	Time Stamps.....	886
<b>Part 8</b>	<b>Macros and Automation.....</b>	<b>889</b>
<b>76</b>	<b>Using Automation and Macros with QlikView.....</b>	<b>890</b>
76.1	The QlikView Automation Interface.....	890
76.2	How Automation and Macros Can Control QlikView.....	890
<b>77</b>	<b>Internal Macro Interpreter.....</b>	<b>893</b>
77.1	Invoking Macros.....	893
77.2	Special Library Functions for JScript.....	894
77.3	Edit Module.....	895
77.4	Macro Debugger.....	897
<b>78</b>	<b>Getting Hold of a QlikView Document.....</b>	<b>899</b>
78.1	Accessing QlikView Documents from the Outside.....	899
78.2	Accessing QV Documents from the Internal Macro Interpreter.....	899
<b>79</b>	<b>VBScript Function Calls from Script.....</b>	<b>901</b>
79.1	Transfer of Parameters.....	902
<b>80</b>	<b>Using Macros in QV Documents on the QV-Server.....</b>	<b>903</b>
80.1	Macros on QlikView Server.....	903
80.2	Macro functionality limitations.....	903
80.3	Macro trigger limitations.....	903
80.4	VBScript functions.....	904
80.5	Server Side Export.....	904
<b>Part 9</b>	<b>Appendix.....</b>	<b>905</b>
<b>A</b>	<b>Limitations and Requirements.....</b>	<b>906</b>
A.1	Limitations of the Amount of Data to be Loaded.....	906
A.2	System Requirements.....	906
A.3	Dynamic Link Libraries (DLLs) Needed.....	906

<b>B Keyboard Command Shortcuts</b> .....	<b>907</b>
B.1 File Menu Command Shortcuts.....	907
B.2 Edit Menu Command Shortcuts.....	907
B.3 View Menu Command Shortcuts.....	908
B.4 Selections Menu Commands Shortcuts.....	908
B.5 Settings Menu Command Shortcuts.....	908
B.6 Bookmarks Menu Command Shortcuts.....	908
B.7 Tools Menu Command Shortcuts.....	908
B.8 Object Menu Command Shortcuts (List box, Statistics box and Open Multi Box).....	909
B.9 Script Keyboard Shortcuts.....	909
B.10 F Key Keyboard Shortcuts.....	909
<b>C Application Performance Optimization</b> .....	<b>911</b>
C.1 Introduction.....	911
C.2 Count (Distinct 'FieldName').....	911
C.3 If ( Condition(Text),.....)	912
C.4 Sum ( If (Condition, 'FieldName' ...)).....	912
C.5 If ( Condition, Sum('FieldName')..)	913
C.6 If ( Condition1, Sum('FieldName'), If (Condition2, Sum('FieldName')).....	913
C.7 Sorting Text.....	914
C.8 Dynamic Captions and Text Objects.....	914
C.9 Macro Triggers (“on change“). .....	915
<b>D Frequently Asked Questions</b> .....	<b>917</b>
D.1 Installation.....	917
D.2 QlikView Documents.....	917
D.3 Scripts and Loading Data.....	918
D.4 QlikView Logic.....	919
D.5 Layout.....	920
D.6 Sharing QlikView Documents with Other People.....	921
<b>E Data Protection Issues</b> .....	<b>923</b>
E.1 Data Security and Integrity.....	923
E.2 The Right to Alter Data.....	923
E.3 The Right to View Data.....	923
E.4 Data Integrity.....	923
<b>E Intentionally Creating Loosely Coupled Tables</b> .....	<b>925</b>
<b>F Backus-Naur Formalism</b> .....	<b>927</b>
<b>Glossary</b> .....	<b>929</b>
<b>Index</b> .....	<b>943</b>



---

# Part 1 Introduction



# 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 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 with an initial capital letter, 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.

## 1.2 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](http://www.qlikview.com) home page (see above).

## 2 What's New in QlikView 11.20?

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 11.20 with 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 General Features

#### Direct Discovery

With QlikView Direct Discovery, business users can leverage any data useful for analysis without scalability limitations. It enables users to perform Business Discovery and visual analysis against any amount of data, regardless of size. With the introduction of this unique hybrid approach, users can associate data stored within Big Data sources directly alongside additional data sources stored within the QlikView in-memory model.

A new script statement, "DIRECT QUERY", is introduced to connect to data in Direct Discovery form. QlikView decides which data resides in-memory and which data is Direct Discovery data by using this special script syntax. This allows certain data elements dictated by the script syntax not to be loaded into the QlikView data model during the script reload process but still be available for query purposes in QlikView objects and to be combined for analysis with the QlikView in-memory dataset.

Once the Direct Discovery structure is established, the Direct Discovery data can be joined with the in-memory data with the common field names. This allows the user to associatively navigate both on the Direct Discovery and in memory data sets.

For more information, see *Direct Discovery (page 161)*.



## 3 Compatibility Issues for Earlier Versions of QlikView

The development team has done absolutely everything we could to make the migration to QlikView 11 from earlier versions as smooth as possible. We believe that we have succeeded and think that the migration from QlikView 7.52, 8, 9 and 10 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, 9, 10 and 11

QlikView 7.52 and later share the same file format as QlikView 11. Work in these versions in parallel virtually without thinking about it. New QlikView 11 document features will of course not work in QlikView 7.52, 8, 9 or 10 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 11 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 11

QlikView 11 cannot save documents in any file format usable in versions prior to QlikView 7.52.

### 3.2 Mixed Client/Server Environments

#### QlikView 11 <-> QlikView 7/8/9/10

QlikView Server 7.52 and later will work seamlessly with QlikView 11 clients as will QlikView Server 11 with QlikView 7.52 and later clients. Of course, in order to make use of QlikView 11 specific functionality you need to have both server and client of version 11. QlikView 11 is not compatible with QlikView versions 7.51 and older.

#### Earlier Versions

QlikView Server 11 is not compatible with QlikView 6 clients. Nor are QlikView Server 6 compatible with QlikView 11 clients.





## 4 QlikView Personal Edition

With QlikView Personal Edition, QlikTech offers a free version of QlikView for personal use. It is meant for individuals, students or small start-ups. QlikView Personal Edition is the full QlikView Desktop product and uses the same installation package. The only difference is that QlikView Personal Edition runs without a license key.

If at any point you or your organization decides to acquire a full QlikView license, this Personal Edition installation will automatically become a full QlikView Desktop without requiring any additional software and without any changes to your QlikView documents.

There are no limitations in terms of time or functionality in QlikView Personal Edition, however there are a few aspects to observe regarding the handling of QlikView documents. In QlikView Personal Edition, every document is saved with a user key that binds that file to the computer it is created on. This implies:

- QlikView Personal Edition can only open files created using that particular copy of QlikView. This means, with QlikView Personal Edition you cannot use your QlikView documents on different computers, you cannot share your QlikView documents with another unregistered user, or open a QlikView document from another user (exempted are documents specially prepared for personal use by QlikTech). However, documents created in QlikView Personal Edition can be used with a licensed copy of QlikView desktop and they can also be published on a QlikView Server.
- You cannot import an entire document layout (sheets/objects/security settings, etc. in XML format, without the data) into QlikView Personal Edition.

Should you change computers, you will not be able to open your previously created documents with QlikView Personal Edition. In this case, you can choose to recover your files. Recovering files means a new user key is generated, assigned to the old document and used for all subsequent files, not only for that particular file. After recovering a file, you can no longer use it on the computer that you created it in.

QlikView will accept 4 user key changes in a document. After that, you will get the error message “Key length has reached its maximum” and the document will not open any more. The only possibility is to recreate it from scratch.

Therefore, you should not recover QlikView documents that you did not create (from a forum or from a colleague, etc.). Doing so will use one of your remaining recovery attempts.

As a user of QlikView Personal edition you can upgrade the software to a newer version at any time and continue working with your documents. New versions may be downloaded on the [QlikView download page](#) and installed on the existing computer.

---

**Note!**

If you have exhausted your recovery attempts, downloading a new version or downloading the same version again will not reset the counter.

---

Users of QlikView Personal Edition do not qualify for telephone or e-mail support from QlikTech, but they can gain excellent QlikView support through our active user community, QlikCommunity. All of the content is accessible to our members, but you must *register* to post questions in the forum, download files or sign up for training. The **Share QlikViews** section of QlikCommunity hosts over countless QlikView applications that have helped our members learn about new, different and even fun uses of QlikView. All these applications have been prepared for use in QlikView Personal Edition.

One of the best ways to learn about the capabilities of QlikView is through the QlikView Tutorial, which provides helpful hints like expression examples, code blocks and design advice. The Tutorial and the corresponding files can be downloaded on the QlikView download page. These documents are also adapted for QlikView Personal Edition.



---

# Part 2 Installation

## 5 Installing QlikView

When you purchase QlikView, you receive an e-mail with the information about the license agreement and with the serial number(s) and the control number(s) needed.

For information about supported operating systems and other system requirements, see *Appendix (page 905)*.

### 5.1 QlikView Installation Packages

You can select between a number of different QlikView installation packages.

The QlikView Desktop, QlikView Server, QlikView WorkBench and QlikView WebParts installations are available in 32-bit and 64-bit versions.

For information about QlikView Server, QlikView WorkBench, QlikView WebParts and QlikView Automation, please refer to the documentation for each of these products.

#### QlikView Desktop Installation Packages

The QlikView Desktop installation packages include the following components:

- The QlikView application
- The QlikView Internet Explorer plugin
- QlikView themes
- QlikView examples
- Help files for all languages that can be selected in the user interface
- The QlikView reference manual in English

You can select **Custom** during the installation progress to reduce the number of installed components.

#### QlikView Documentation and Tutorial Packages

You can download and install QlikView documentation separately. Please note that most of the information in the QlikView reference manual can be found in the help files, that are automatically included in the standard QlikView installation. The help files are available in all languages that can be selected in the user interface.

The complete documentation and tutorial installation package includes the following components:

- The QlikView Server reference manual
- The QlikView WorkBench reference manual
- The QlikView WebParts reference manual
- The Qlik View Automation reference manual
- The QlikView reference manual in all languages that can be selected in the user interface
- The QlikView tutorial in all languages that can be selected in the user interface
- The API guide (a QlikView document that describes the COM API)

The complete QlikView Desktop documentation and tutorial installation package includes the following components:

- The QlikView reference manual in all languages that can be selected in the user interface
- The QlikView tutorial in all languages that can be selected in the user interface
- The API guide (a QlikView document that describes the COM API)

The QlikView Desktop documentation and tutorial installation package in a certain language includes the following components:

- The QlikView reference manual in the selected language
- The QlikView tutorial in the selected language

## 5.2 Downloading and Starting the QlikView Desktop Installation Program

Do the following:

1. Go to [www.qlikview.com/download](http://www.qlikview.com/download) and select a QlikView Desktop installation package.
2. Register or login to start the download of QlikView.
3. Double-click the Setup.exe file to start the installation.

The installation program should now have started.

## 5.3 When the Installation Program Is Running

1. First, select the desired installation language from the drop down list, and click **OK**.
2. Wait while the installation is prepared, according to the dialogs displayed.
3. When the **Welcome** dialog opens, read the information and click **Next**.
4. The software license agreement opens. Read it, and click **I accept the license agreement** (if this is the case), then click **Next**.
5. In the **Customer Information** dialog you can specify for whom the installation will be personalized. Click **Next** to continue.
6. If you prefer another destination folder than the default one for the program installation, specify this by clicking **Change** in the **Destination Folder** dialog. Then click **Next**.
7. The **Setup Type** dialog appears. If **Complete** is chosen, the following features will be installed to the path specified in the previous dialog: The QlikView program and its help file, the QlikView Internet Explorer Plug-In, and examples of QlikView functionality and features. If **Custom** is chosen, a dialog where you can specify which features to install will appear when you click **Next**. Make the selections and click **Next**.
8. The installation is now ready to start. Click **Install** to start it.
9. In the last dialog, click **Finish** to complete the installation process.

---

### Note!

Always enter the QlikView license on the account that installed it; otherwise it might not function properly.

---

## Logging the Installation

When Setup.exe is run, a log file is written to the temp folder of the user. The log file is called QlikViewx86.wil for the x86 version and QlikViewx64.wil for the 64-bit version. Each time the installation is run a new file is generated, over writing the old log file.

## QlikView Settings File

As of version 10, all settings for QlikView are saved in a file instead of written to the registry. The file, Settings.ini, is located in C:\Users\username\AppData\Roaming\QlikTech\QlikView on Windows Vista, Windows 7 and Windows 8. On older systems the file is found in C:\Documents and Settings\username\Application Data\QlikTech\QlikView.

The settings for QlikView OCX are found in the same file.

## 5.4 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** page. 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.

## 5.5 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.

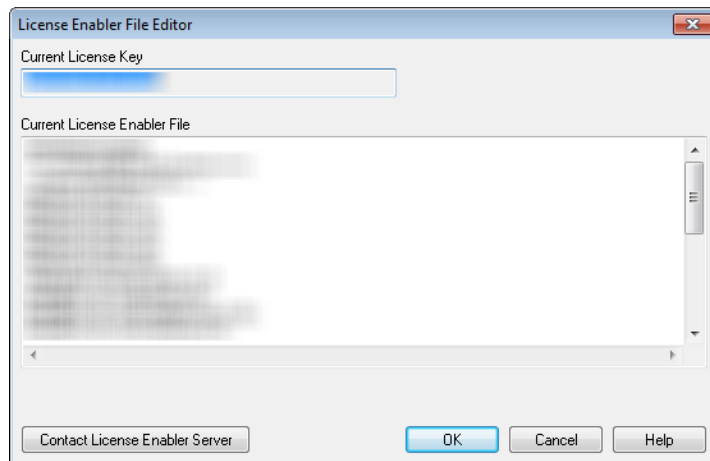
To distribute a QlikView installation, you need the executable installation file. Serial numbers and control numbers are not necessary until you start your first QlikView session on the client computer.

QlikView 11 uses the Microsoft Windows Installer technique (MSI packaging).

To find out more about how to install an MSI-package, please see the Windows installer pages on <http://msdn.microsoft.com>.

## 5.6 License Enabler File Editor

You can also update your license and review the License Enabler File by selecting **License Update** from the **Help** menu. This requires no network access.

*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** page. It is also possible to run Qlikview without a bought license key. In this case, a license lease from a QlikView Server - a Named CAL - or a Personal Edition of QlikView is needed.

### 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 QlikTech 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.

## 5.7 Updating QlikView

In the **Help** menu in QlikView, you find the option **QlikView Update...**. Choosing this will open the 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.





## 6 OLE DB and ODBC

### 6.1 OLE DB

QlikView supports the OLE DB interface for connection to external data sources. A great number of external databases can be accessed via the Microsoft OLE DB Provider for ODBC Drivers.

### 6.2 When Is the ODBC Interface Needed?

To access a general database, it is necessary to have the ODBC (Open DataBase Connectivity) interface installed. 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 the DBMS manufacturer. Some drivers are redistributed freely.

The ODBC interface described here is the interface on the client computer. If the plan is to use ODBC to access a multi-user relational database on a network server, additional DBMS software that allows a client to access the database on the server might be needed. Contact the 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. The 32-bit version of QlikView will only work with the 32-bit ODBC drivers. However, the 64-bit version will work with both 32-bit and 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**.

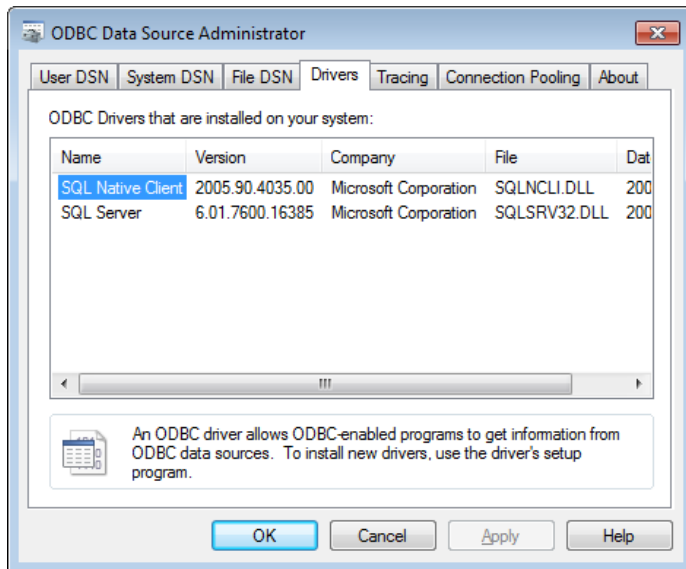
When using the 32-bit versions of ODBC and QlikView, you must start the 32-bit administrator via the executable file, `odbcad32.exe`, that is located in the SysWOW64 folder, usually `c:\windows\SysWOW64`.

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 the system. (For 32-bit ODBC on a 64-bit operating system start `odbcad32.exe`).

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



*The ODBC Data Source Administrator dialog, Drivers tab.*

If the driver needed is not found on the ODBC Drivers tab, contact the software supplier to obtain the correct driver.

## 6.3 Creating ODBC Data Sources

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

Before the start of creating data sources, a decision must be made whether they should be user or system data sources. User data sources cannot be reached unless the correct user id was used when logging on. To share the data sources with other users, system data sources must be created.

1. Open the **ODBC Data Source Administrator** dialog again.
2. Go to the tab **User DSN** to create user data sources, or to the tab **System DSN** to create system data sources.
3. Click the **Add** button. The **Add Data Source** dialog 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 specific to the selected database driver will appear. Name the data source and set the necessary parameters. Click the **OK** button when ready.

---

# Part 3 Navigation and Basics

## 7 Starting QlikView

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

To start QlikView from the command line, use the parameters described in the following section.

### 7.1 Command Line Syntax

The QlikView command line invocation can be described by the following syntax (for a description of the syntax notation used, see *Backus-Naur Formalism (page 927)*):

*[ 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 the 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 (page 36)*.

**/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 *Error Variables (page 288)*)

---

### 7.2 The Start Page

The start page contains several sections listed in the table below.

**Examples**

A set of examples that you can explore in order to learn how to use QlikView in various ways.

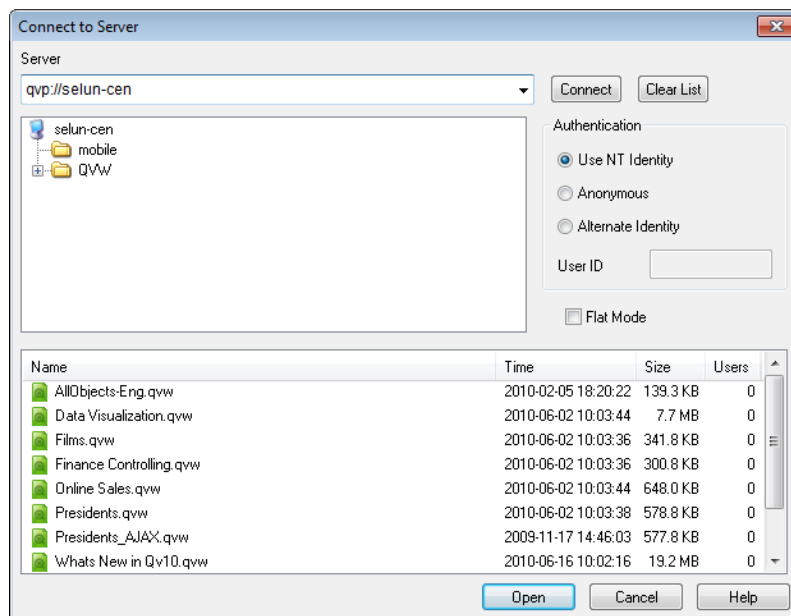
- 
- Recent** Contains a list of recently opened documents and web pages. Just click on one of them to re-open. Right-click on a document to gain access to a menu with the following useful commands:
- Open "document" Without Data**  
Opens the document but skips table and field data. A layout with all sheets and sheet objects in place but empty is opened. 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 the script needs to be re-run to re-populate the document after the changes). This is done with the command line switch **nodata**( *Command Line Syntax (page 36)*) as well.
  - 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 the 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.
- Favorites** Contains a list of the documents and web pages previously added to the favorites list. Unlike the list of recently used documents the favorites list remains unchanged until explicitly adding or deleting entries. Just click on a favorite document or page to re-open. Right-click on a document to gain access to a menu with the following useful commands:
- Open "document" Without Data**  
Opens the document but skips variables and table and field data. A layout with all sheets and sheet objects in place but empty is opened. 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 the script needs to be re-run to re-populate the document after the changes). This can also be accomplished with the command line switch **nodata**.
  - 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.
- Additional learning** Contains guides and other resources for learning more about QlikView, such as tutorials, online training and QlikView Community.

Click the **New document** button if you want to create a new document.

QlikView will remember which tab was last used and open the start page on that tab the next time that QlikView starts. Deselect the **Show Start Page When Launching QlikView** check box, if the start page are not to appear when the program starts.

The Start page can be left open and will then be found as **Start Page** on the **Windows** menu. If the start page is closed, it can be reopened at any time by choosing **Show Start Page** from the **Help** menu.

## 7.3 Connect to Server



### *The Connect to Server dialog*

This dialog is used for opening QlikView documents remotely on a QlikView Server.

Note that when opening documents as a client to QlikView Server there will be restriction in what is possible to do with the document compared to when opening local documents. As a client it might not be possible to add or delete sheet objects, depending on whether the document supports collaboration. Sheets cannot be added or deleted. Furthermore, most properties dialogs and the script cannot be accessed, macros cannot be changed and the document cannot be saved.

Enter the computer name or the URL to a computer running QlikView Server under **Server**. Then click on **Connect**. Recently visited servers can be retrieved from the drop-down list. A URL may contain additional connection data according to the *Connection pseudo-URLs: (page 39)* described at the bottom of this help topic.

Once connected, the central pane of the dialog displays the document folders available on the QlikView Server. Click on a folder and the available documents will be shown under **Name** at the bottom of the dialog. Double-click a document or select it and click on **Open** in order to open the document.

### **Clear List**

The button clears the list of **Recent Servers** on the **Start Page**.

**Authentication**

Select how to identify yourself to the QlikView Server.  
The following three authentication methods are available:

**Use NT Identity**

Uses authenticated NT identity. This only works if the computer is connected to the same Windows NT domain as the QlikView Server.

**Anonymous**

Uses anonymous login. It is only possible to connect if the QlikView Server is set to allow anonymous users and it is only possible to see documents with file permissions for the QlikView Server anonymous account. The connection pseudo-URL under **Server** will be complemented by a @ before the server name.

**Alternate Identity**

Uses a specific NT user name known on the QlikView Server's domain. The user name is typed in the **User ID** edit box. It will prompt for the corresponding user password when pressing the **Connect** button. The connection pseudo-URL under **Server** will be complemented by *username@* before the server name.

**Flat Mode**

With this setting enabled, the bottom pane of the dialog will display the contents of all subfolders simultaneously rather than using the tree-view approach.

**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 (excluding qvw extension). The parameter is optional.

*port* (e.g. 4749) can be used for specifying a specific port used by the server

*protocol* (e.g. http) can be used for specifying a tunneling protocol. The parameter is optional.

@ without *username* denotes anonymous identity.

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

## 7.4 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 the end-users with up-to-date information and letting you manage the 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 there is no access to QlikView Server or QlikView Publisher, it is possible to reload QlikView documents on the stand-alone application. Then use the command line switches described in *Command Line Syntax (page 36)*.

## Scheduling Programs

Windows has a **Task Scheduler** service. The services are controlled from the Administrative Tools (Windows 7 and Windows 8), 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 access to network resources is needed, reconfigure the service to run under a different account. Since the QlikView license is stored in the account-specific folder, make sure that it is correctly registered for the account to be used. To do this 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 it is possible to 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 (page 417)*.

## ODBC

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



Control Panel.



## 8 Getting Started Wizard

The Getting Started Wizard helps create a QlikView document by loading data and creating a chart in the following steps:

Step 1 - Select a data source	page 43
Step 2 - Data presentation	page 43
Step 3 - Save file	page 43
Step 4 - Choose a chart type	page 43
Step 5 - Populate your chart	page 44
Step 6 - Add object for making selections	page 44

### Starting the Getting Started Wizard

The Getting Started Wizard is initiated when a new QlikView document starts.

On the first page of the wizard it is possible to control whether to show the wizard or not when creating a new QlikView document. This setting can also be made in the **User Preferences** dialog that is reached from the **Settings** menu.

#### Step 1 - Select a data source

The wizard can only handle Excel files. **Browse** to the file containing the data. Note that only the first work sheet in an Excel file is loaded. If the data is in non-Excel format, use the **Script Editor** to select the data.

Click **Insert data** if you want to borrow some data from us.

Click **Next step** to continue.

#### Step 2 - Data presentation

Verify the presentation of the data and select whether to use the first row in the Excel sheet as headers or enter a new one. Column headers are also called **field names**.

To use predefined headers from the data source, select **Use column headers from data file**. To create new column headers, select **Add column headers**. Each column will get a heading in the format A, B, etcetera. To enter new headings, click the heading (for example A) and type the new heading. Click Enter to leave the header.

Click **Next step** to continue.

#### Step 3 - Save file

In the **Save As** dialog, browse to the folder where the QlikView file is to be saved and enter a file name.

Click **Save** to close the dialog.

To enter or change the file path click the **Save As...** button to re-open the **Save As** dialog.

Click **Next step** to continue and create a chart.

#### Step 4 - Choose a chart type

Select type of chart to be created by clicking the corresponding icon. The chart types available are those most commonly used in QlikView. It is possible to go back to the chart and change it into any other QlikView chart type via the **Chart Properties** dialog after finishing the wizard.

Click **Next step** to continue.

**Note!**

In this step the **Back** button is disabled.

---

## Step 5 - Populate your chart

### Dimension

Select a dimension in the drop-down box.

Dimensions define the values for which the chart expressions will be calculated. Dimensions are typically found to the left in table charts and on the x-axis in for example bar charts.

### Expression

Chart expressions define the calculated values in the chart. Expressions are typically found to the right in table charts and on the y-axis in for example bar charts.

Expressions in QlikView can range from short and simple to long and complex calculations. This step makes it possible to choose between three very common expressions.

**Calculate the sum of:**

Choose this option to see the numeric sum of a field, for example sum(Sales). Then choose which field to sum up in the drop down list.

**Calculate the average for:**

Choose this option to see the numeric average (mean) of a field, for example avg(Score). Then choose a field for the calculation in the drop down list.

**Count the number of:**

Choose this option to see the number of values in a field, for example count(OrderID). Then choose in which field to count the values in the drop down list.

Click **Create a second chart** if you want to create another chart. This will take you back to step 4.

Click **Next step** to continue.

## Step 6 - Add object for making selections

Choose the type of object you want to use for making selections.

**Listboxes:**

Select the fields you want to create listboxes for. You can select up to five fields in this wizard, but it is possible to add more once you have created the document.

**Table box:**

This option will automatically include all the available fields and place them in a table.

Click **Create** to close the wizard and create the document.

After finishing this wizard it is possible to add or change more dimensions and expressions at any time via the **Chart Properties** dialog. The chart properties dialog is reached by right-clicking on the chart and selecting **Properties...**

## 9 QlikView File

A QlikView document is a file containing everything needed 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.

### 9.1 QlikView Project Files

It is possible to save a QlikView document into several files that can be used for versioning. Each file defines a property of the document, a sheet, an object, the script etc.

Each time the document is opened and an object or a setting is changed, these changes are saved to the different files, making it easy to follow the changes made in the document. This way it is also possible to see who made a change and to which part of the document.

To create these project files it is necessary to create a folder next to the qvw file with the same name as the QlikView document and add -prj, e.g. the project folder for a document called Finance.qvw should be Finance-prj.

---

**Note!**

No data from the document will be saved in the project files.

---

#### Files in the Project Folder

The file `QlikView.txt` contains a list of all the objects part of the QlikView document.

The different sheets and objects in the list are named after their object ID. The project files are:

- `QlikView.txt` - contains a list of all the files included in the project
- `AllProperties.xml`
- `DocProperties.xml`
- `DocInternals.xml`
- `TopLayout.xml`
- `LoadScript.txt` - contains the document load script
- `Module.txt` - contains the document macro code, if any exists
- `Module.txt` - a single file is created for every sheet. In addition the files contain references to all the sheet objects on the sheet.
- Separate files are also created for every sheet object:
  - `LB<id>.xml`
  - `SB<id>.xml`
  - `MB<id>.xml`
  - `TB<id>.xml`
  - `CH<id>.xml`
  - `IB<id>.xml`
  - `CS<id>.xml`
  - `BU<id>.xml`
  - `TX<id>.xml`
  - `LA<id>.xml`
  - `SL<id>.xml`

- SO<id>.xml
- BM<id>.xml
- CT<id>.xml
- RP<id>.xml

The files `DocProperties.xml`, `AllProperties.xml`, `DocInternals.xml` and `TopLayout.xml` all contain property settings for the different parts of the document. `DocBinary.dat` contains user sensitive data, such as passwords.

## 9.2 Source Control

It is possible to connect QlikView Desktop to a source control system; the ‘QvMsscciProvider.exe’ and ‘QvSvnProvider.exe’ are the providers used for this giving you source control function in QlikView 11 SR1 or later to both Microsoft Team Foundation Server and Subversion.

Once connected to a source control system, QlikView developers can add projects to source control. During the **Add** process the:

- QlikView document is saved
- Project folder is created
- Project files are exported into the project folder
- Project files are added to source control
- Project settings file is created

The project settings file is stored in the local project folder and contains the settings necessary to access the source control information for the project. The project settings file is not included in the files managed by the source control system. The existence of the project settings file tells QlikView that a given document is managed by a source control system. In addition, QlikView has to be able to find and load the needed Provider DLL, in order to treat the document as attached to source control. This check is performed every time a document is opened in QlikView and in QV11 SR1 or later a settings file appears in the project folder with information about the Source Control Provider, i.e. `SourceControlSettings.ini`. The only exception relates to backward compatibility issues. If a project folder ‘abc-prj’ does not contain ‘`SourceControlSettings.ini`’ but contains a file called `abc-prj.scc` it means it was created using the V11 Initial Release (not SR1 or later) and we should use the Msscci Source Control Provider to open the project.

If the document is attached to source control, the status bar will contain an indication of the document’s status.

QlikView integration with source control only affects document layout. No actual data loaded into QlikView is placed into nor fetched from, source control. The operation, **Get Project from Source Control** will load a document that contains everything except data. A Reload must be executed to populate the document with data.

After a QlikView document is connected to source control, saving the document will automatically check out the files that have changes. QlikView does not provide any other way of checking out the project files. If Source Control system is not available while performing document Save, QlikView will try to work "off-line", i.e. remove read only flags from the modified files and save the latest version. On the next document save, when Source Control system is available, QlikView will perform check out for all locally modified files.

On document save, project files are not checked in. To check in document updates to source control, the **Check In Pending Changes** menu item must be used. A single QlikView check in operation can generate several changesets when using Microsoft TFS and only one when using Subversion. As a result, rollback between randomly selected changesets in Microsoft TFS does not guarantee that the document will be in a stable state.






For more information, refer to the Source Control white paper that is available at [community.qlikview.com](http://community.qlikview.com).





# 10 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 *Customize (Toolbars)... (page 64)* for more information.

## 10.1 File Menu

The **File** menu is a drop-down menu at the top of the screen, and contains the following commands:

<b>New</b>	Opens a new QlikView window and allows to create a new <i>QlikView File (page 45)</i> .	
<b>Open...</b>	Opens a new QlikView window and allows to open a QlikView file or a Table file. Opening a table file automatically opens the <i>File Wizard: Type (page 198)</i> . This command can also be invoked by the following keyboard shortcut: Ctrl+O.	
<b>Open in Server...</b>	Opens the <i>Connect to Server (page 38)</i> . In this dialog it is possible to connect to a QlikView Server and browse for a document to open in client mode. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+O.	
<hr/>		
<b>Note!</b>		
When opening documents as a client to QlikView Server the possibilities are restricted on what is possible to do with the document compared to when opening local documents. As a client it is not possible to add or delete sheet objects, depending on whether the document supports collaboration. Sheets cannot be added or deleted. Furthermore, not all properties dialogs can be accessed, macros changed, the script accessed, data reloaded or the document saved.		
<hr/>		
<b>Refresh Document</b>	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 access is gained to the latest data, while maintaining the session including selections and layout state.	
<b>Open URL...</b>	Opens the <b>Open URL</b> dialog. In this dialog 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 for, for example, the QlikView Publisher Accesspoint or for pages displaying QlikView Server documents via AJAX clients. Opened web pages can be accessed via the <b>Windows</b> menu just as with standard QlikView document windows.	
<b>Open FTP...</b>	Opens a new QlikView window, allows the opening of a QlikView file or table file from an Ftp server, (see <i>Open Internet Files or Open QlikView Document (page 193)</i> ). Opening a table file automatically opens the <b>File Wizard</b> .	
<b>Close</b>	Closes the active QlikView file.	
<b>Favorites</b>	This cascade menu controls a list of user-defined favorite documents, managed independently of the <b>Recent documents</b> list (see below).	
<b>Save</b>	Saves the present configuration in a QlikView file. Data, script and layout are saved. This command can also be invoked by the following keyboard shortcut: Ctrl+S. If you want to share a QlikView document using AJAX clients, the file name must not contain hash (#) characters.	

<b>Save As...</b>	Saves the present configuration in a new QlikView file under a new name. This command can also be invoked by the following keyboard shortcut: F12. If you want to share a QlikView document using AJAX clients, the file name must not contain hash (#) characters.	
<b>Save Link...</b>	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.	
<b>Mail as Attachment...</b>	Only available when working with a local document. Creates a 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). For this command to work a mail client has to be configured on the computer.	
<b>Mail with Bookmark as a Link...</b>	Only available when working with a QlikView Server document. Creates a 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, obviously provided that he has access rights to the document and data therein. A mail client has to be configured on the computer for this command to work. The QlikView Server must be configured to allow server bookmarks	
<b>Print...</b>	Opens the standard <b>Print</b> dialog (see <i>Exporting and Printing (page 91)</i> ), allows to print the current sheet object. This command is not available for list boxes. This command can also be invoked by the following keyboard shortcut: Ctrl+P.	
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+P.	
<b>Print Possible...</b>	Opens the standard <b>Print</b> dialog (see <i>Exporting and Printing (page 91)</i> ), allows to print the possible (selected and optional) values of the current list box. This command is only available for list boxes.	
<b>Print Possible as PDF...</b>	This command must be used when printing list boxes and multi boxes as PDF.	
<b>Print Sheet...</b>	Opens the <b>Print</b> dialog, allows to print an image of the current sheet, see <i>Print Sheet (page 95)</i> .	
<b>Print Preview...</b>	Shows preview of printable sheet objects, see <i>Print Preview (page 96)</i> .	



**Source Control****Settings**

Set which MSSCCI (Microsoft Source Code Control Interface) provider dll to use when adding a project or getting a project from source control. For more information see *Source Control (page 46)*.

**Add Project to Source Control**

Saves the opened QlikView document and automatically creates a project folder and project files. Invokes the source control system to add the project to source control.

**Get Project from Source Control**

Retrieves a project from source control and rebuilds the QlikView document from the project files. QlikView data is not stored in source control. To fill the document with the data the command Reload has to be executed on the rebuilt document.

**Get Latest Version**

Retrieves the latest version of the document from the source control system and rebuilds the QlikView document.

**Check In Pending Changes**






Saves the QlikView document and performs a check in operation to the source control system. This can include four different source control operations with their own dialogs:

- Add newly created files.
- Undo the checked out files that need to be deleted from the source control.
- Remove the files that correspond to the removed sheets or sheet objects that are part of the source control.
- Check in modified files.

If the document was not up to date before the Check In Pending Changes operation, then reopen the document to load the project changes that could be introduced while resolving the conflicts.

**Undo Pending Changes**








Undoes the changes. If there are locally modified files that are not checked out in the Source Control system, those files are checked out before the undo check-out operation is performed. This operation will rebuild the QlikView document by deleting files that were added and not checked in, and references to pending delete files are restored.

<b>Export</b>	<p><b>Export Contents...</b> This alternative is only available when an exportable sheet object is active. It then performs the export operation found on the sheet object's <b>Object</b> menu.</p> <p><b>Export Sheet Image...</b> Opens a dialog for saving an image of the current sheet to file.</p> <p><b>Export Document Layout</b> Opens a dialog for saving the document layout as an XML file. No data from the document are saved in the XML file.</p> <p><b>Export Sheet Layout...</b> 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.</p>	
<b>Import</b>	<p><b>Import Document Layout</b> Opens a dialog for importing the document layout of another document saved as XML. No data are imported.</p> <p><b>Import Sheet Layout</b> Opens a dialog for importing the layout from another sheet saved as XML. No data are imported.</p>	
<b>Edit Script...</b>	Opens the <i>Edit Script Dialog</i> (page 175). In this it is possible to write and execute scripts that open and connect to databases.	
<b>Reload</b>	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 by the following keyboard shortcut: Ctrl+R.	
<b>Partial Reload</b>	Executes the current load script, including all script commands, such as <i>Drop Table</i> (page 237), and reloads data to the active QlikView document. However, only those tables whose load and select statements are preceded by the <i>Replace</i> (page 268) or <i>Add</i> (page 222) 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 by the following keyboard shortcut: Ctrl+Shift+R.	
<b>Reduce Data</b>	Opens a menu containing the following two commands: <p><b>Keep Possible Values</b> Reduces the QlikView database by removing all excluded values.</p> <p><b>Remove All Values</b> Creates a template by removing all the values from the QlikView database, but keeping the database structure and the layout.</p>	
<b>Table Viewer...</b>	Opens the <i>Table Viewer</i> (page 183) 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 by the following keyboard shortcut: Ctrl+T.	
<b>Recent documents list</b>	A listing of the last QlikView documents used. The number of documents shown can be set in the <b>Settings: User Preferences</b> dialog. Default number is 8. Selecting one of these is equivalent to the <b>Open</b> command.	

**Exit** Closes the open documents and exits QlikView.

## 10.2 Edit Menu

The **Edit** menu is a drop-down menu at the top of the screen, and contains the following commands:

<b>Undo Layout Change</b>	Undoes the latest layout change, including moving, sizing and removing of sheet objects as well as changes to sheet object properties. Removing sheets, editing the sheet properties or document properties can also be undone. This command can also be invoked by the following keyboard shortcut: Ctrl+Z.	
<b>Redo Layout Change</b>	Redoes the latest undone layout action. This command can also be invoked by the following keyboard shortcut: Ctrl+Y.	
<b>Cut</b>	Moves the selected sheet object(s) to <b>Clipboard</b> for pasting elsewhere in the QlikView document. If only one sheet object is active a bitmap image of the object, which can be pasted into other programs, will also be put into <b>Clipboard</b> . This command can also be invoked by the following keyboard shortcut: Ctrl+X.	
<b>Copy</b>	Copies the selected sheet object(s) to <b>Clipboard</b> for pasting elsewhere in the QlikView document. If only one sheet object is active a bitmap image of the object, which can be pasted into other programs, will also be put into <b>Clipboard</b> . This command can also be invoked by the following keyboard shortcut: Ctrl+C.	
<b>Paste</b>	Pastes one or several sheet objects from <b>Clipboard</b> back to the QlikView document. This command can also be invoked by the following keyboard shortcut: Ctrl+V.	
<b>Format Painter</b>	The <i>Format Painter</i> (page 63) tool makes it possible to copy formatting from one sheet object to another.	
<b>Remove</b>	If the <b>Confirm Remove Sheet Object</b> command in the <i>User Preferences: Objects</i> (page 78) dialog is enabled (default) the user will be asked to confirm this command before deleting the sheet object(s).	
<b>Activate All</b>	Makes every sheet object on the sheet active. Another way of obtaining a similar result is to click and drag a rectangle around those sheet objects that should be activated, or to Shift-click them. This command can also be invoked by the following keyboard shortcut: Ctrl+A.	
<b>Search</b>	Opens the Text search box, provided that a searchable sheet object (list box or opened multi box) is active. This command can also be invoked by the following keyboard shortcut: Ctrl+F.	
<b>Fuzzy Search</b>	Opens the Text search box in <i>Fuzzy Search</i> (page 107), provided that a searchable sheet object is active.	
<b>Advanced Search</b>	Opens the <i>Advanced Search Dialog</i> (page 108), 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 by the following keyboard shortcut: Ctrl+Shift+F.	

**Copy mode** Switches from logic mode to copy mode. Values clicked while in copy mode are copied to **Clipboard** without changing the logical state of the QlikView run. When copy mode is chosen, the *Clipboard Copy List (page 96)* dialog opens. In this dialog the copied fields are shown. The format of the copy list can also be set here.




## 10.3 View Menu

The **View** menu is a drop-down menu at the top of the screen, and contains the following commands:







<b>Sheets</b>	Upon selection, a cascade menu displays a list of all sheets used in the document, sorted as they appear from left to right.
<b>Toolbars</b>	<p>Upon selection a cascade menu, where it is possible to enable any toolbar to be displayed with the document, opens. The following toolbars can all be employed independently of each other: <b>Standard Toolbar</b>, <b>Navigation Toolbar</b>, <b>Design Toolbar</b>, <b>Sheets Toolbar</b> and <b>Bookmarks Toolbar</b>. The last entry in the list is the <i>Customize (Toolbars)... (page 64)</i> option allows the configuration of the toolbar according to the users own preferences.</p> <p>The <b>Standard</b>, <b>Navigation</b> and <b>Design</b> toolbars have each been predefined with a suitable selection of menu commands, as hinted by the names. The <b>Sheet</b> and <b>Bookmark</b> toolbars are actually defined as drop-down lists, offering an optional means for navigation between sheets and bookmarks, respectively. Note that the contents and functionality of any toolbar can be configured in a number of ways via the <b>Customize</b> option.</p> <p><u>Tip!</u> To the far left in all toolbars and also in the menu bar, a symbol of vertical dots is displayed. Clicking and dragging here un-docks or docks the bar in the standard Windows fashion.</p>
<b>Statusbar</b>	Toggles the status bar on or off.
<b>Zoom</b>	Permits zooming in 25% increments of the sheet area to better fit different screen resolutions. Other <b>Zoom</b> factors can be specified in the <i>Sheet Properties: General (page 463)</i> page.
<b>Server Objects</b>	Toggles the <i>Server Objects Pane (page 599)</i> pane on and off.
<b>Resize Window</b>	With this option the window size of the QlikView document can be set to one of several common screen resolution.
<b>Fit Zoom to Window</b>	Alters the size of the displayed active sheet and everything on it so that it fits the window size.
<b>Apply Zoom to all Sheets</b>	The <b>Zoom</b> settings of the displayed active sheet will be transferred to all sheets of the document.
<b>Design Grid</b>	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 by the following keyboard short-cut: Ctrl+G.
<b>Turn on/off WebView mode</b>	Toggles WebView mode, which uses the internal web browser in QlikView to display the document in AJAX mode, on and off.



<b>Current Selections...</b>	Opens the <i>Current Selections (page 103)</i> dialog where selections are listed by field name and field value. This command can also be invoked by the following keyboard shortcut: Ctrl+O.	
------------------------------	---	---





## 10.4 Selections Menu




The **Selections** menu is a drop-down menu at the top of the screen, and contains the following commands:

<b>Back</b>	Reverts to the preceding logical state. Applies to value selections and all the commands in the <b>Selections</b> menu. QlikView maintains a list of the 100 latest states. Each <b>Back</b> will take one step further back in the list. This command can also be invoked by the following keyboard shortcut: Shift+Left arrow.	
<b>Forward</b>	Reverts to the logical state before a <b>Back</b> command. Toggle between two states by alternating between the <b>Back</b> and <b>Forward</b> commands. This command can also be invoked by the following keyboard shortcut: Shift + Right arrow.	
<b>Lock</b>	Locks all the current value selections. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+L.	
<b>Unlock</b>	Unlocks all the currently locked value selections. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+U.	
<b>Clear</b>	Applies the start selection of a QlikView document, which can be configured, see <b>Set Clear State</b> below. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+D.	
<b>Clear All</b>	Clears all the current selections except the ones that are locked.	
<b>Unlock and Clear All</b>	Clears all the current value selections.	
<b>Set Clear State</b>	Sets the current selection as <b>Clear State</b> .	
<b>Reset Clear State</b>	Resets <b>Clear State</b> .	

## 10.5 Layout Menu





The **Layout** menu is a drop-down menu at the top of the screen, and contains the following commands:

<b>Add Sheet...</b>	Adds a tabbed sheet in which a new set of sheet objects can be displayed. Provided that both client and server are QlikView version 9 or later, it is possible to add a new sheet to the layout even when working with a document on a QlikView Server.	
<b>Promote Sheet</b>	Moves the (tab of the) current sheet one step to the left (forwards).	
<b>Demote Sheet</b>	Moves the (tab of the) current sheet one step to the right (back).	
<b>Remove Sheet</b>	Deletes the active sheet and everything on it.	

<b>Select Fields...</b>	Opens the <b>Fields</b> property page of the <b>Sheet Properties</b> sheet. One or several field names can be selected from a list. The selected fields will be displayed in default configured list boxes on the active sheet. To set the specific properties of a list box right click on it and select <b>Properties</b> from the float menu (or from <b>Object</b> in the Main Menu). This command is not available when working with documents on QlikView Server.	
<b>New Sheet Object</b>	Opens a cascade menu in which it is possible to choose to create one of the different sheet objects. When created, the corresponding Properties dialog will be presented, for configuring the new object. Provided that both client and server are QlikView version 8 or later it may be possible to add new sheet objects to the layout even if working with a document on QlikView Server. This requires <ul style="list-style-type: none"> <li>a) that the license is valid,</li> <li>b) that the server document has been configured to allow server objects and</li> <li>c) that the QlikView Server is configured to allow server objects.</li> </ul>	
<b>Rearrange Sheet Objects</b>	Automatically arranges the sheet objects on the active sheet.	
<b>Adjust Off-Screen Objects</b>	Any sheet object outside the visible area of the QlikView window is repositioned back inside the visible area.	
<b>Align/Distribute</b>	The alignment of sheet objects can be determined here according to a number of alternatives e.g. <b>Left Align, Center Horizontally</b> .	

## 10.6 Settings Menu



The **Settings** menu is a drop-down menu at the top of the screen, containing the following commands:

<b>User Preferences...</b>	Opens the <i>User Preferences (page 69)</i> dialog that contains settings that the user does not normally change when switching to another document. This command can also be invoked by the following keyboard shortcut: Ctrl+Alt+U.	
<b>Document Properties...</b>	Opens the <i>Document Properties: General (page 424)</i> dialog that contains settings for the entire document. Some common properties for the sheet objects in the document can also be set in this dialog. This command can also be invoked by the following keyboard shortcut: Ctrl+Alt+D.	
<b>Sheet Properties...</b>	Opens the <i>Sheet Properties: General (page 463)</i> dialog that contains settings for 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 by the following keyboard shortcut: Ctrl+Alt+S. See the links at the bottom of this page.	
<b>Variable Overview...</b>	Opens the <i>Variable Overview (page 153)</i> dialog where all non-hidden variables and their values are shown in a single list. This command can also be invoked by the following keyboard shortcut: Ctrl+Alt+V.	

**Expression Overview...** Opens the *Expression Overview (page 155)* dialog where all document, sheet and sheet object expressions can be shown and centrally maintained in a single list. This command can also be invoked by the following keyboard shortcut: Ctrl+Alt+E.

## 10.7 Bookmarks Menu

The **Bookmarks** menu is a drop-down menu at the top of the screen, and contains the following commands:

<b>Document Bookmarks</b>	The first ten document bookmarks of the active document can be retrieved from this list.	
<b>User Bookmarks</b>	The first ten personal bookmarks linked to the active document can be retrieved from this list.	
<b>Add Bookmark</b>	Opens the <i>Add Bookmark Dialog (page 117)</i> dialog, where the bookmark name can be edited. This command can also be invoked by the following keyboard shortcut: Ctrl+B.	
<b>Replace Bookmark</b>	The first ten document bookmarks are listed above the first ten personal bookmarks of the active document. The command replaces the selection state of the selected bookmark with the current state.	
<b>Remove Bookmark</b>	The first ten document bookmarks are listed above the first ten personal bookmarks of the active document. The command removes the selected bookmark.	
<b>More...</b>	Opens the <b>Bookmarks</b> dialog, where all previously created bookmarks for the document can be retrieved.	
<b>Import...</b>	After browsing for and selecting a previously saved bookmark (.qbm) file, the <i>Import Bookmarks (page 122)</i> dialog will open to make it possible to import bookmarks.	
<b>Export...</b>	Opens the <i>Export Bookmark(s) (page 121)</i> dialog where selected bookmarks can be exported to a QlikView bookmark (.qbm) file.	


## 10.8 Reports Menu




The **Reports** menu at the top of the QlikView screen contains the command **Edit Reports...** that opens the *Report Editor (page 126)* dialog where new reports can be created or existing reports edited. From this dialog it is also possible to delete and select reports, design their layout, add pages and images etcetera. The report editor lists all available reports for immediate selection.

Reports may be **Document Reports**, stored with the document, or **User Reports**, stored separately on the user's computer. See *Report Editor (page 126)*.

## 10.9 Tools Menu

The **Tools** menu is a drop-down menu at the top of the screen, and contains the following commands:

<b>Edit module...</b>	Opens the <i>Edit Module (page 895)</i> dialog used for creating macros.	
-----------------------	--	---

<b>Open QlikView AccessPoint</b>	Opens the QlikView AccessPoint in a html window inside QlikView. This command is only available if a URL for QlikView AccessPoint has been configured on the <b>User Preferences: Locations</b> page.	
<b>Open QlikView Management Console</b>	Opens the QlikView Management Console/QlikView Enterprise Management Console in an html window inside QlikView. This command is only available if a URL for QlikView Management Console/QlikView Enterprise Management Console has been configured on the <b>User Preferences: Locations</b> page.	
<b>Quick Chart Wizard...</b>	Opens the <i>Quick Chart Wizard (page 767)</i> , which helps creating a simple chart in a quick and easy way, without bothering about the great number of different settings and options available.	
<b>Time Chart Wizard...</b>	The <i>Time Chart Wizard (page 775)</i> helps with the common task of building 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 etcetera.	
<b>Statistics Chart Wizard...</b>	The <i>Statistics Chart Wizard (page 781)</i> provides guidance for those who want to apply common statistical tests on data in QlikView.	
<b>Box Plot Wizard</b>	The <i>Boxplot Wizard (page 765)</i> creates a combo chart, often used for display of statistical data.	
<b>Alerts...</b>	Opens the <i>Alerts (page 142)</i> dialog where alerts can be defined and edited.	
<b>Alert Wizard...</b>	The <i>Alert Wizard (page 146)</i> helps with the task of defining an alert.	
<b>Theme Maker Wizard</b>	Invokes the <i>Theme Maker Wizard (page 603)</i> for creating a new theme or editing an existing theme. More details about themes can be found under <i>Layout Themes (page 601)</i> .	

## 10.10 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.

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.

If there is more than one active sheet object, the Object menu contains the commands that are common to the active sheet objects.

For a description of the different object menus, see the section on the specific object.

## 10.11 Window Menu

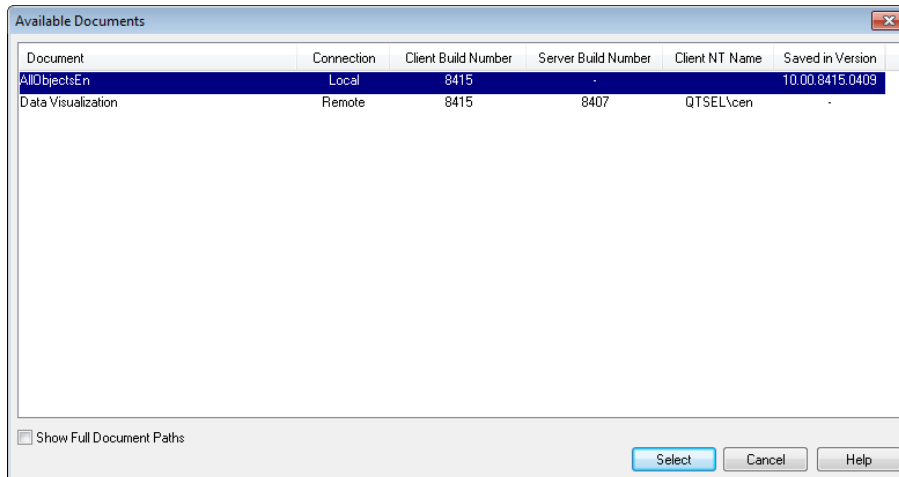
The **Window** menu is a drop-down menu at the top of the screen, containing the following commands:

<b>Cascade</b>	Will arrange windows so that they overlap.
<b>Tile</b>	Will arrange windows as non-overlapping tiles.
<b>Arrange Icons</b>	Will arrange icons at the bottom of the window.



**Available Documents** Opens the *Available Documents* (page 57) dialog, displaying an expanded listing (beyond 10) of open QlikView documents and providing additional document information. Selecting a document makes it the active window.

## Available Documents



This dialog can be used for switching the active document when more than ten documents are open at the same time. Select a document in the list and click on **Select** to make that document the active window.

This dialog also contains a wealth of support information that should be reported whenever problems are experienced with a specific QlikView document. The columns are:

<b>Document</b>	The name of the .qvw document. If <b>Show Full Document Paths</b> at the bottom of the dialog is checked, the document name will be given with its full file path.
<b>Connection</b>	<b>Local</b> (document opened on local computer) or <b>Remote</b> (document opened on QlikView Server).
<b>Client Build Number</b>	The build number of the QlikView client.
<b>Server Build Number</b>	The build number of QlikView Server for remote documents.
<b>Client NT Name</b>	The Windows NT authenticated identity of client user when connection is made with NT authentication.
<b>Saved in Version</b>	Full information about the QlikView version that was used the last time the document was saved (available for local documents only).

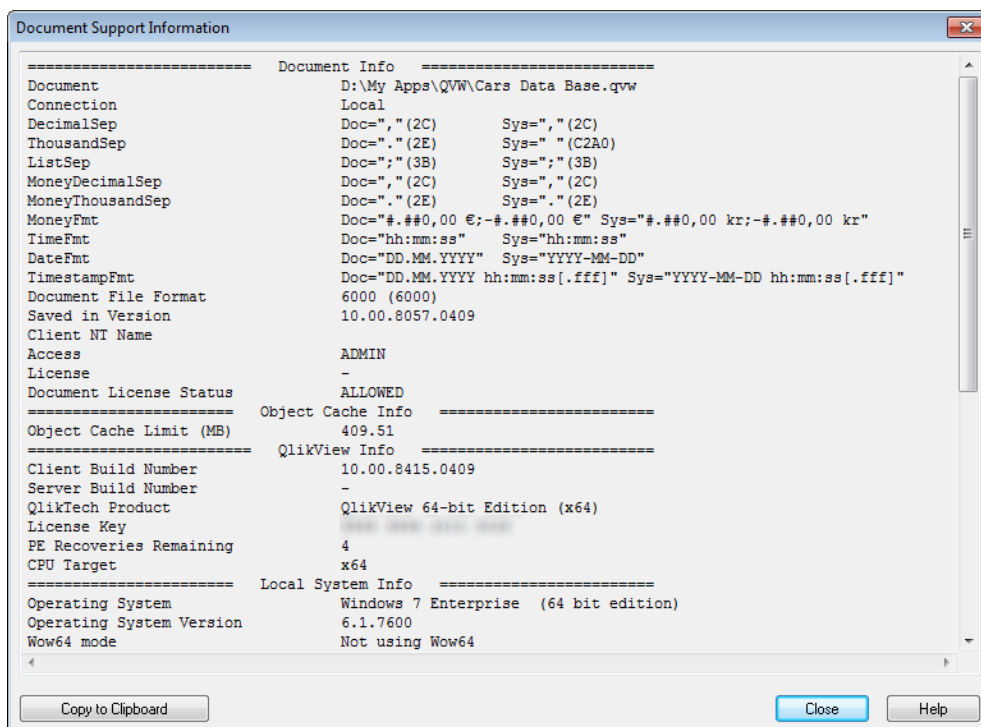
## 10.12 Help Menu

The **Help** menu is a drop-down menu at the top of the screen, containing the following commands:

<b>Contents</b>	Starts the QlikView help.
<b>Using Help</b>	Gives some useful advice on how to use the QlikView help.
<b>Show Start Page</b>	Shows the start page each time QlikView starts. If the start page should not appear when the program starts, deselect the <b>Show Start Page When Launching QlikView</b> check box.

<b>QlikTech on the Web</b>	Opens up further options that redirects to the QlikTech home page.
<b>QlikView Update...</b>	Opens the <i>QlikView Update</i> dialog from which it is possible to make QlikView contact the QlikView update server to see if any program updates are available.
<b>License Update...</b>	Opens the <i>License Enabler File Editor</i> dialog where the user can review or edit the License Enabler File.
<b>Document Support Info...</b>	Opens the <i>Document Support Information (page 58)</i> dialog, displaying a list of support data pertaining to the active document.
<b>About QlikView...</b>	Opens the <i>About</i> dialog which displays the QlikView version, serial number and owner name.

## Document Support Information



This dialog displays a list of support data pertaining to the active document. This information should be referenced whenever bugs or problems that may be related to specific QlikView documents are reported. Note that some rows are only relevant for local documents while others refer only to remote documents.

Via the **Copy to Clipboard** button it is possible to copy and paste the contents of this dialog into e.g. a text document attachment or directly into the support e-mail at the users convenience.

# 11 Toolbars and Status Bar

## 11.1 Toolbars

QlikView has five standard toolbars and one menubar. The standard toolbar normally contains buttons for tasks performed while using QlikView documents, whereas the design toolbar normally contains buttons for tasks performed 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 enabled and disabled. 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. The toolbars can be docked to any side of the QlikView application window.

## 11.2 Standard Toolbar



### The standard toolbar

The QlikView standard toolbar 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.

<b>New File</b>	Opens a new QlikView window and allows to create a new <i>QlikView File (page 45)</i> . This command can also be invoked by the following keyboard shortcut: Ctrl+N.	
<b>Open File</b>	Allows the opening of a <i>QlikView File (page 45)</i> or <i>Table file</i> in a new QlikView window. Opening a table file automatically opens the <i>File Wizard: Type (page 198)</i> . This command can also be invoked by the following keyboard shortcut: Ctrl+O.	
<b>Refresh</b>	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 access is gained to the latest data, while maintaining the session including selections and layout state.	
<b>Save</b>	Saves the active document as a file. The default file format is set in <i>User Preferences: Save (page 73)</i> .	
<b>Print...</b>	Clicking the print tool immediately effects the printout of the selected sheet object according to default printer settings, bypassing the <i>Print: General (page 91)</i> property page. The tool will be grayed if no printable object is selected. This command can also be invoked by the following keyboard shortcut: Ctrl+P.	

<b>Print as PDF</b>	<p>Opens the <i>Print: General (page 91)</i> with the QlikViewPDF printer pre-selected.</p> <p>After clicking the <b>Print</b> button 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 QlikView's homepage and installed separately.</p>	
<b>Edit Script</b>	<p>Opens the <i>Edit Script Dialog (page 175)</i> dialog, making it possible to write and execute <i>Script</i> that open databases and retrieve data into QlikView. This command can also be invoked by the following keyboard shortcut: Ctrl+E.</p>	
<b>Reload</b>	<p>Re-executes the current <i>Script</i>, updating the associative QlikView database to include changes made in source data since the previous execution. The time of the latest reload is shown as a timestamp in the status bar.</p>	
<b>Undo Layout Change</b>	<p>Undoes the latest layout change, including moving, sizing and removing of sheet objects as well as changes to sheet object properties. Removing sheets, editing the sheet properties or document properties can also be undone.</p> <p>QlikView maintains a list of the latest layout changes. Each <b>Undo Layout Change</b> command makes it take one step backward in the list. Certain operations, e.g. <b>Reload</b> and <b>Reduce Data</b> will empty the <b>Undo/Redo</b> buffer. This command can also be invoked by the following keyboard shortcut: Ctrl+Z.</p>	
<b>Redo Layout Change</b>	<p>Redoes the latest undone layout action. Each <b>Redo Layout Change</b> command makes it take one step forward in the list, as long as there are undone actions to redo.</p> <p>Certain operations, e.g. <b>Reload</b> and <b>Reduce Data</b> will empty the <b>Undo/Redo</b> buffer. This command can also be invoked by the following keyboard shortcut: Ctrl+Y.</p>	
<b>Search</b>	<p>Opens the <i>Search (page 105)</i> box for the active object. This command can also be invoked by the following keyboard shortcut: Ctrl+F.</p>	
<b>Current Selections</b>	<p>Opens the <b>Current Selections</b> dialog in which it is possible to see the selections that are active. This command can also be invoked by the following keyboard shortcut: Ctrl+O.</p>	
<b>Quick Chart Wizard</b>	<p>Opens the <i>Quick Chart Wizard (page 767)</i> it allows to create a chart in a quick and easy way, without bothering about the great number of different settings and options available.</p>	
<b>Add Bookmark</b>	<p>Opens the <i>Add Bookmark Dialog (page 117)</i> dialog, where the bookmark name can be edited. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+B.</p>	
<b>Help Topics</b>	<p>Opens the QlikView help.</p>	
<b>Context Help</b>	<p>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 help is needed.</p>	

## 11.3 Navigation Toolbar



The QlikView navigation toolbar 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 text below refers 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 by the following keyboard shortcut: Ctrl+Shift+D.

**Clear All**

Clears all selections, excluding locked ones.

**Unlock and Clear all**

Unlocks and clears all selections.

**Clear Specific State**

Clears the selections of a specific state.

**Set Clear State**

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

**Reset Clear State**

Resets **Clear State** to no selections.

**Back** Reverts to the preceding logical (selection) state. Applies to value selections and all the commands in the **Edit** menu except **Copy** and **Copy Mode**. QlikView maintains a list of the 100 latest states. Each **Back** command will take one step further back in the list.

**Forward** Reverts to the logical state before a **Back** command. It is possible to toggle between two states by alternating between the **Back** and **Forward** commands.

**Lock (Selections)** Locks all the current *value selections* in the entire document.

**Unlock (Selections)** Unlocks all the currently locked *value selections* in the entire document.

## 11.4 Design Toolbar



The QlikView design toolbar (see above) contains buttons for tasks to 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.



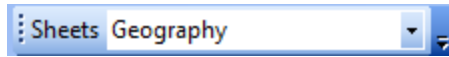
---

<b>Demote Sheet</b>	Moves the active sheet one step further to the right.	
<b>Sheet Properties</b>	Opens the <b>Sheet Properties</b> dialog, from which it is possible to modify the active sheet.	
<b>Create List Box</b>	Creates a list box to display the field of choice from the database table.	
<b>Create Statistics Box</b>	Creates a statistics box, which calculates statistical entities based on the possible values of a field.	
<b>Create Table Box</b>	Creates a table box, suitable for showing record-oriented information.	
<b>Create Multi Box</b>	Creates a multi box, suitable for showing different attributes.	
<b>Create Chart</b>	Creates a chart that can be made to display fields and calculated dimensions.	
<b>Create Input Box</b>	Creates an input box, suitable for displaying and entering data into QlikView variables.	
<b>Create Current Selections Box</b>	Creates a current selections box, suitable for displaying the current selections directly in the layout.	
<b>Create Button</b>	Creates a button object that performs actions in QlikView, e.g. shortcut, export etc.	
<b>Create Text Object</b>	Creates an object for displaying text information or images.	
<b>Create Line/Arrow Object</b>	Creates a line/arrow object, suitable for drawing a line or an arrow in the layout.	
<b>Create Slider/Calendar Object</b>	Creates a new slider/calendar object.	
<b>Create Bookmark Object</b>	Creates a new bookmark object.	
<b>Create Search Object</b>	Creates a new search object.	
<b>Create Container</b>	Creates a new container.	
<b>Create Custom Object</b>	Creates a new custom object.	
<b>Create Time Chart</b>	The <b>time chart wizard</b> helps 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.	

---

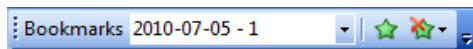
<b>Format Painter</b>	This button makes it possible to copy formatting from one sheet objects to one or many other sheet objects. In order to copy formatting to a single object, first click on the source object, then single-click on the format painter button and then click on the target object. In order 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. Stop the copying by clicking the button again or pressing Esc. When copying formatting between sheet objects of different types or when the user click on the caption of the target object(s), only border/caption properties will be copied. When copying between sheet objects of the same type, additional object type specific properties will be copied.	
<b>Align Left</b>	Aligns the active sheet objects along their left border.	
<b>Center Horizontally</b>	Aligns the active sheet objects along their center on the horizontal axis.	
<b>Align Right</b>	Aligns the active sheet objects along their right border.	
<b>Align Bottom</b>	Aligns the active sheet objects along their bottom border.	
<b>Center Vertically</b>	Aligns the active sheet objects along their center on the vertical axis.	
<b>Align Top</b>	Aligns the active sheet objects along their top border.	
<b>Space Horizontally</b>	Distributes the active sheet objects on the horizontal axis with equal spaces between them.	
<b>Space Vertically</b>	Distributes the active sheet objects on the vertical axis with equal spaces between them.	
<b>Adjust Left</b>	Arranges the active sheet objects from the vertical edge of the left-most object and to the right with minimal spaces between them.	
<b>Adjust Top</b>	Arranges the active sheet objects from the horizontal top edge of the top-most object and downwards with minimal spaces between them.	
<b>Document Properties</b>	Opens the <b>Document Properties</b> dialog, from which it is possible to modify the settings of the current document.	
<b>User Preferences</b>	Opens the <b>User Preferences</b> dialog, where it is possible to modify settings concerning the way the user works.	
<b>Edit Module</b>	Opens the Edit Module dialog where macros and custom defined functions can be written in VBScript or JScript.	
<b>Table Viewer</b>	Opens the Table Viewer dialog where the data table structure is displayed.	
<b>WebView Mode</b>	Toggles WebView mode, which uses the internal web browser in Qlik-View to display the document layout as an AJAX page.	

## 11.5 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.

## 11.6 Bookmarks Toolbar



The bookmarks 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. 

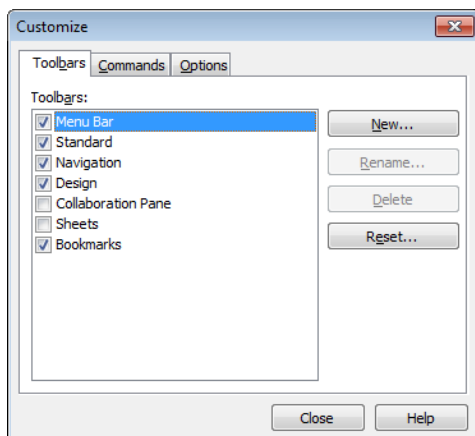
## 11.7 Customize (Toolbars)...

The **Customize** dialog allows users to configure toolbars according to their own preferences. (While this is generally a very practical functionality, there are situations where the document designer may wish to enforce a standard setting to be applied for all users of the document. See *Customizing QlikView Toolbars in Large Deployments* (page 66).)

There are actually two types of customizable toolbars; the ones which are already defined (i.e. named) in QlikView, and the ones which can be completely custom defined as needed.

The **Customize** dialog consists of the three tabs, **Toolbars**, **Commands** and **Options** described below:

### Toolbars



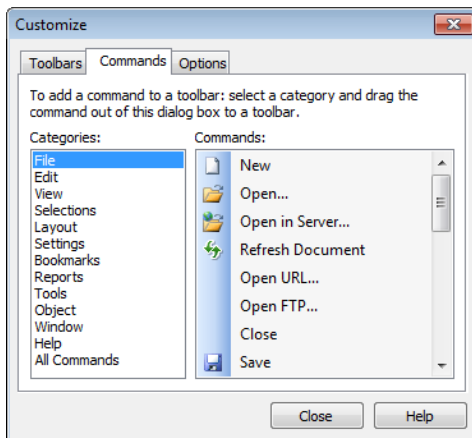
*Customize dialog, Toolbars*

This page contains a list of all available toolbars and the menu bar. Activate/deactivate toolbars by ticking the relevant check box in the list.



<b>New</b>	Creates a new toolbar.
<b>Rename</b>	Renames the highlighted toolbar. This command is not available for the five default toolbars.
<b>Delete</b>	Deletes the highlighted toolbar. This command is not available for the five default toolbars.
<b>Reset</b>	Resets the configuration of the highlighted toolbar to default settings.

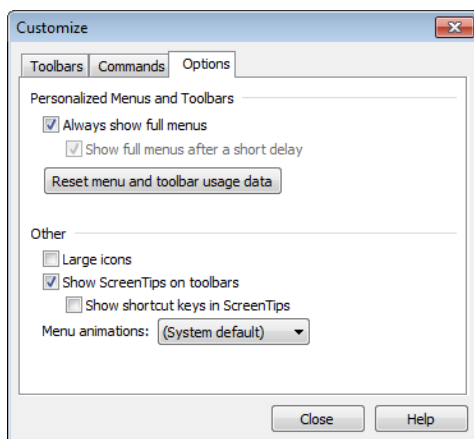
## Commands



*Customize dialog, Commands*

As the instruction in the **Commands** tab states, adding to or deducting from the contents of any toolbar is simply a matter of dragging any menu command from the **Commands** pane to the desired toolbar or vice versa. First the toolbar being modified must of course be set to display mode in the **Toolbars** tab. By means of the **Categories** pane it is possible to limit the listed commands to the contents of a single QlikView menu at a time.

## Options



*Customize dialog, Options*

The **Options** tab contains additional settings for customization:

In the **Personalized Menus and Toolbars** section it is possible to set if shorter menus with only the most common commands should be used.

<b>Always show full menus</b>	Disable this option to use shorter menus with only the most common commands. <b>Show full menus after a short delay</b> When short menus are used, enable this setting to have the full menu appear after hovering over the menu.
<b>Reset menu and toolbar usage data</b>	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 following display options have been grouped under **Other**:

<b>Large icons</b>	This setting draws the toolbar icons at double size, improving visibility if needed.
<b>Show Screen Tips on toolbars</b>	This option toggles the tool-tips on or off.
<b>Show shortcut keys in Screen Tips</b>	This option adds to the information displayed in the tool-tips pop-up.
<b>Menu animations:</b>	This setting applies to the way menus (and cascade menus) open. In addition to the <b>System Default</b> setting, a number of custom animations are available in a drop-down list.

## 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 settings in Settings.ini.

The ini file is found in *C:\Users\username\AppData\Roaming\QlikTech\productname* on Windows Vista, Windows 7 and Windows 8. On older systems the file is found in *C:\Documents and Settings\username\Application Data\QlikTech\productname*.

*AllowCustomizeToolbars*

and

*AllowCustomizeMenubar*

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

---

#### **Note!**

Before changing these settings, make sure the application is closed.

---

### Enforcing Toolbar Settings on Large Numbers of Computers

To duplicate the toolbar settings between computers:

1. Customize the toolbars on one computer.
2. Copy the file Settings.ini.

---

#### **Note!**

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

---

## 11.8 Status Bar

Below the sheet, the status bar is found. Choose **View** and **Status bar** 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.

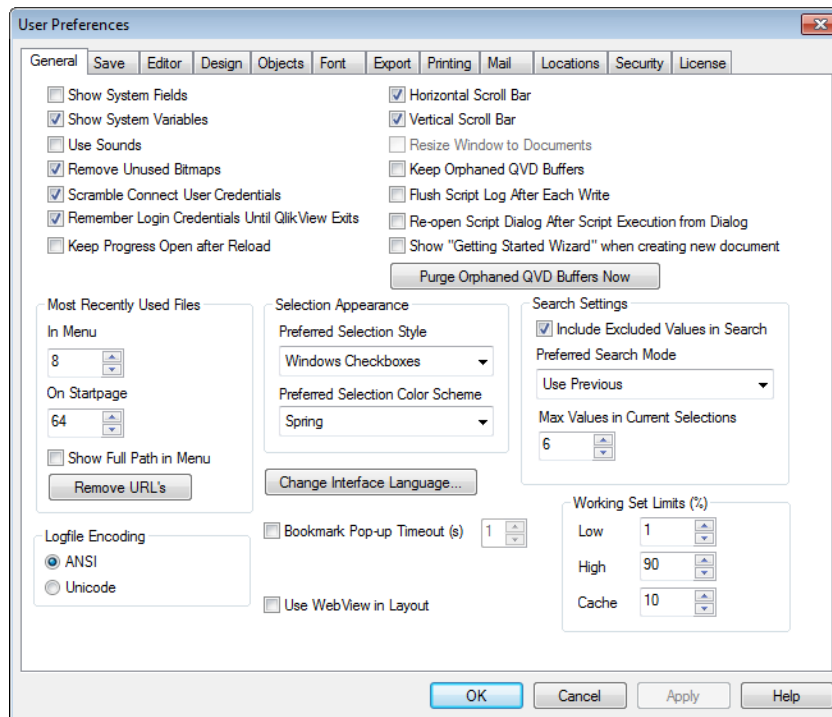
Finally a selection indicator is shown on the status bar. It will be green if selections are made that cannot be seen on the current sheet.



# 12 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.

## 12.1 User Preferences: General



### User Preferences, General

- |  |  |
|--|--|
| <b>Show System Fields</b>                              | If selected, the <i>System Fields</i> (page 379) will be included by default in all listings of field names.   |
| <b>Show System Variables</b>                           | If selected, the <b>System Variables</b> will be included by default in all listings of variables.   |
| <b>Use Sounds</b>                                      | Enables the built in sound effects of QlikView.  |
| <b>Remove Unused Bitmaps</b>                           | Bitmaps used in sheets and sheet objects are normally retained in the document even if their use is turned off. Check this alternative to have them automatically removed.   |
| <b>Scramble Connect User Credentials</b>               | Check this alternative if the connect statement wizard should mask the <b>connect</b> statements by scrambling user ID and password.   |
| <b>Remember Login Credentials until QlikView exits</b> | QlikView can cache the UserID and Password for any QlikView-documents requiring login that have been opened for the duration of a QlikView-session. By deselecting this option, the user will be prompted to login each time a document is reopened. |

<b>Keep Progress Open after Reload</b>	Checking this box will keep the <b>Script Execution Progress</b> dialog open after the script has finished reloading.
<b>Horizontal Scroll bar</b>	If this setting is on, a horizontal scroll bar will appear along the edge of the sheet whenever the positioning or sizing of sheet objects cannot be fully accommodated horizontally inside the application window.
<b>Vertical Scroll Bar</b>	As the above setting, but pertaining to the vertical.
<b>Resize Window to Documents</b>	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 at the time the document was saved.
<b>Keep Orphaned QVD Buffers</b>	This setting overrules the normal procedure whereby any automatically created <i>QVD Files (page 409)</i> are purged immediately after script execution, provided that the document that created them is no longer available. This setting should normally be left unchecked.
<b>Flush script log after each write</b>	For performance reasons, the script log is normally not written to disk after each individual statement. By selecting this script box it will be. This may be useful if other programs are used 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.
<b>Re-open Script Dialog After Script Execution from Dialog</b>	If the script is executed from within the <b>Edit Script</b> dialog and this check box is marked, the dialog will be re-opened after script execution.
<b>Show Getting Started Wizard When Creating New Document</b>	Opens the <b>Getting Started Wizard</b> which guides through the steps of creating a new document based on a single Excel sheet.
<b>Purge Orphaned QVD Buffers Now</b>	Click this button to perform a manual purge of orphaned QVD buffers that have been retained by the setting <b>Keep Orphaned QVD buffers</b> above.
<b>Most Recently Used Files</b>	<p>This group is used for controlling the list of recently opened QlikView files in the <b>File</b> menu and on the <b>Start Page</b>.</p> <p>Under <b>In Menu</b> the number of recently used files listed in the <b>File</b> menu can be changed. The default is 8.</p> <p>Under <b>On Startpage</b> the number of recently used files listed on the <b>Start Page</b> can be changed. The default is 64.</p> <p>These changes take effect after restart.</p> <p>If the alternative <b>Show Full Path in Menu</b> is checked, the <b>File</b> menu will be widened as necessary to show full paths in the list of recently used files.</p> <p>Clicking the <b>Remove URLs</b> button will remove any URLs from the list of most recently used files.</p>
<b>Logfile Encoding</b>	In this group it is possible to set the character set for the log file. The default setting is <b>ANSI</b> , but it is possible to check the option <b>Unicode</b> instead.

- Selection Appearance**      **Preferred Selection Style**  
Sets the default selection style. 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 **Led Checkboxes** styles all use color coding for indication of selected, possible and excluded values. The **Windows Checkboxes** style and the **LED Checkboxes** style mimic the standard Windows interface with a check box at each value. This default can be overridden for a specific document by means of a corresponding setting under *Document Properties: General (page 424)*. There are limitations as to the behavior of certain macro triggers when working with documents on QlikView Server. More information about the selection styles can be found in the section *Selection Styles (page 99)*.
- Preferred Selection Color Scheme**  
Sets the default selection color scheme. When using the selection styles based on color, there are a number of different color schemes available. The *Color Codes (page 99)* (green for selected, blue for locked etc.) cannot be changed, but variations of tone and intensity are possible. This default can be overridden for a specific document by means of a corresponding setting under *Document Properties: General (page 424)*.
- Change Interface Language**      Press the button to open the **Select Interface Language** dialog. In this dialog all language versions of QlikView that are available on the computer will be listed. (Various language versions are included as .dll files at the time of installation.) After selecting a new language, QlikView must be restarted to effect the change. The change affects both the language of the program user interface and the language of the online help, provided that the appropriate help files are also available on the computer.  
The interface language can also be set by modifying the `settings.ini` file, found in `C:\Users\username\AppData\Roaming\QlikTech\QlikView`. Edit the setting `InterfaceLanguage` and set it to a supported language.
- Bookmark Popup Timeout (s)**      Check this alternative and enter a time delay to have bookmark pop-up windows automatically close after a set number of seconds.
- Use WebView in Layout**      Toggles **WebView** mode, which uses the internal web browser in QlikView to display the document layout as an AJAX page, on and off.

**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: it is possible to search either among optional values, or among all values, i.e. include the excluded values in the search. Enable this option 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 the user starts typing directly and do not use any of the menu choices or keyboard shortcuts for starting the search. It is possible to override this setting by selecting a different setting on sheet object level.

**Use Previous**

The search mode of the last completed search will be used.

**Use Wildcard Search**

The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search.

**Use Fuzzy Search**

The initial search string will be a tilde (~) to denote a fuzzy search.

**Use Normal Search**

No additional characters will be added to the search string. Without wildcards, a Normal Search will be made.

**Max Values in Current Selections**

Specify the maximum number of distinct selected values here 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 Qlik-View.

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.

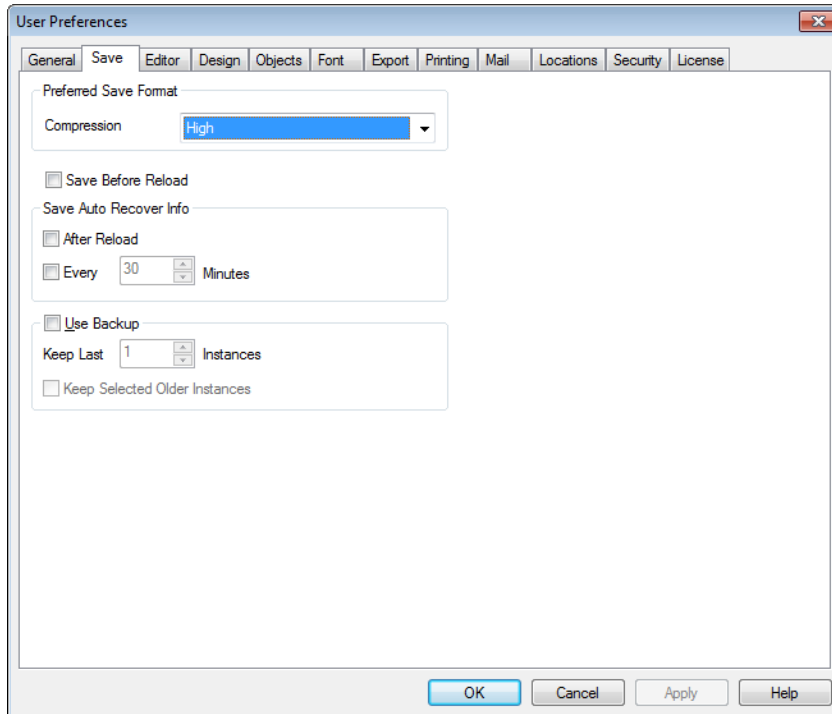
**Cache**

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



## 12.2 User Preferences: Save

This dialog tab contains settings for how QlikView documents should be saved.



*User Preferences, Save*

**Preferred Save Format** Here a default save format can be set for all new files. To make changes that affect the current document only, select a **Save Format** under *Document Properties: General* (page 424) instead.

### Compression

This drop-down 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 are saved without compression.

**Save Before Reload** Check this alternative to have the document automatically saved before the script is executed.

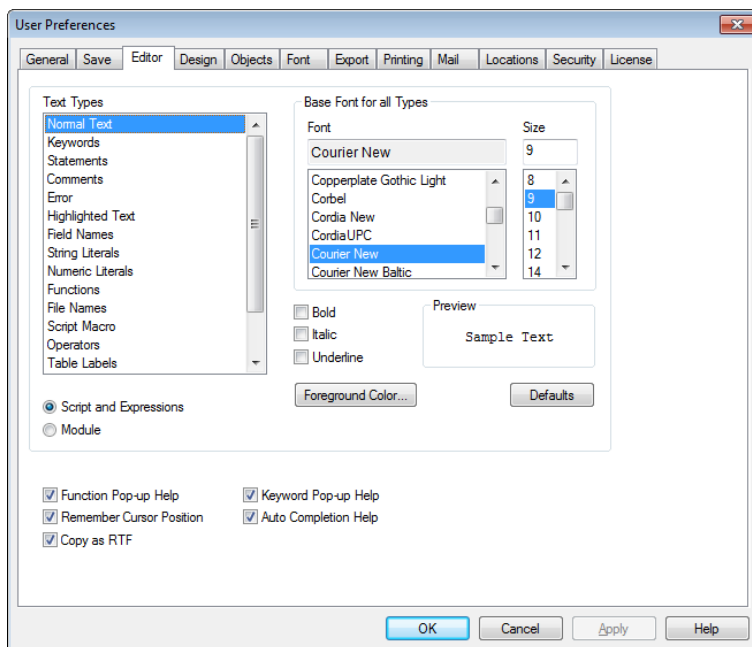
**Save AutoRecover Info** Here rules for creating a backup of the current QlikView file can be specified. This autosave functionality can be very helpful in case of a system crash. It is possible to independently set the auto recover save to occur at regular intervals (**Every \_ Minutes**) and each time the script is run (**After Reload**).

**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** Here it is possible to specify rules for keeping earlier copies that have been created by the autosave functionality. With **Use Backup** checked, it is possible to specify how many backup versions will be retained (**Keep Last \_ Instances**) and the program may also retain a selection of preferred older versions (**Keep Selected Older Instances**).

## 12.3 User Preferences: Editor



### *User Preferences, Editor*

In this dialog it is possible to determine the personal preferences for text in the QlikView editors. These include the editors in the *Edit Script Dialog* (page 175) (**Script**), the *Edit Expression Dialog* (page 757) dialog (**Expression in Chart**) and the macro editor in the *Edit Module* (page 895) dialog (**Module**). Here it is possible to set different preferences in font, font size, font style and font color for a number of different **Text Types** that can be highlighted in the text box.

## Script and Expressions

<b>Normal Text</b>	Text that does not belong to any of the categories described below.
<b>Keywords</b>	The keywords used in the script (described under <i>Script Statements and Keywords (page 219)</i> ), e.g. load, select, directory, semantic, etc.
<b>Comments</b>	Comments typed in the script or chart.
<b>Error</b>	The errors that QlikView detects in the script or chart expression.
<b>Highlighted Text</b>	When approaching 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.
<b>Field Names</b>	The names of the fields to be loaded and used in charts.
<b>Literals</b>	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 <i>Quotation Marks in Scripting (page 292)</i> .
<b>Functions</b>	The functions used in the script (described under <i>Aggregation Functions (page 298)</i> ), e.g. div, left, if, num, etc. and charts.
<b>File Names</b>	The name of the file from which the fields are to be retrieved.
<b>Script Macro</b>	The variables used in the script. For more information on variables, see <i>Script Variables (page 282)</i> .
<b>Aggregation Functions</b>	The aggregations functions used in the expression (described under <i>Aggregation Functions (page 298)</i> ), e.g. sum, min, max, etc.
<b>Operators</b>	The operators used in the script (described under <i>Operators (page 787)</i> ), e.g. +, like, etc. and in expressions.
<b>Table Labels</b>	The labels assigned to specific tables. For more information see <i>Table Names (page 281)</i> .
<b>Set Analysis</b>	The set analysis identifiers, modifiers and operators used in the expression (described under <i>Set Analysis (page 819)</i> ).

## Module

<b>Normal Text</b>	Text that does not belong to any of the categories described below.
<b>Keywords</b>	Visual Basic Script reserved keywords.
<b>Highlighted Text</b>	When approaching 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

<b>Normal Text</b>	Text that does not belong to any of the categories described below.
<b>Comments</b>	Comments typed in the chart expression.
<b>Error</b>	The errors that QlikView detects in the expression, e.g. a missing parenthesis.
<b>Highlighted Text</b>	When approaching 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 easily be detected.
<b>Field Names</b>	The names of the fields used.
<b>Functions</b>	QlikView standard function names for chart expressions.
<b>File Name</b>	The name of the file from which the fields are to be retrieved.
<b>Aggregation Functions</b>	The aggregations functions used in the expression (described under <i>Aggregation Functions (page 298)</i> ), e.g. sum, min, max, etc.
<b>Operators</b>	The operators used in the expression (described under <i>Operators (page 787)</i> ), e. g. +, like, etc.
<b>Set Analysis</b>	The set analysis identifiers, modifiers and operators used in the expression (described under <i>Set Analysis (page 819)</i> ).

### 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 bold, italic and underline can be set individually using the check boxes.

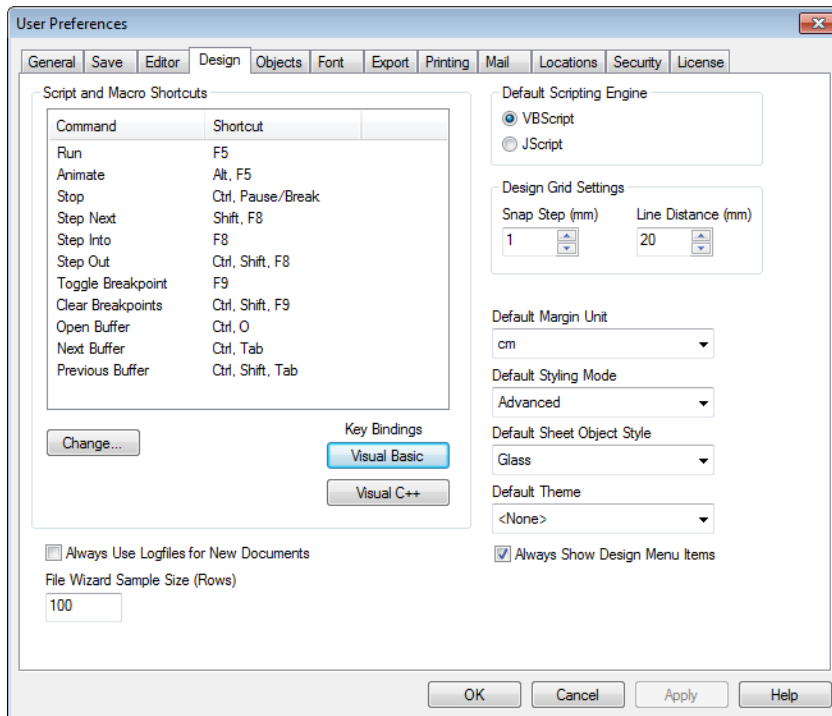
Clicking on the **Foreground Color** button brings up a color palette where it is possible to pick a predefined color or define a custom foreground color.

To reset values, click the **Defaults** button.

The following features may also be used in editors:

<b>Function Popup Help</b>	Check this alternative to have a small pop-up window appear when typing in a function.
<b>Remember Cursor Position</b>	Check this alternative to have the editors for script and macro remember the position of the cursor when leaving the editor. 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.
<b>Copy as RTF</b>	Check this alternative to have text copied from the editor windows stored in the <b>Clipboard</b> not only as pure text but also as RTF. This makes it possible to paste the text with full formatting into applications supporting RTF import.

## 12.4 User Preferences: Design



### *User Preferences, Design*

In the **Script and Macro Shortcuts** group it is possible to 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 click the **Change** button to customize the keyboard shortcut for the command.

There are two optional **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.

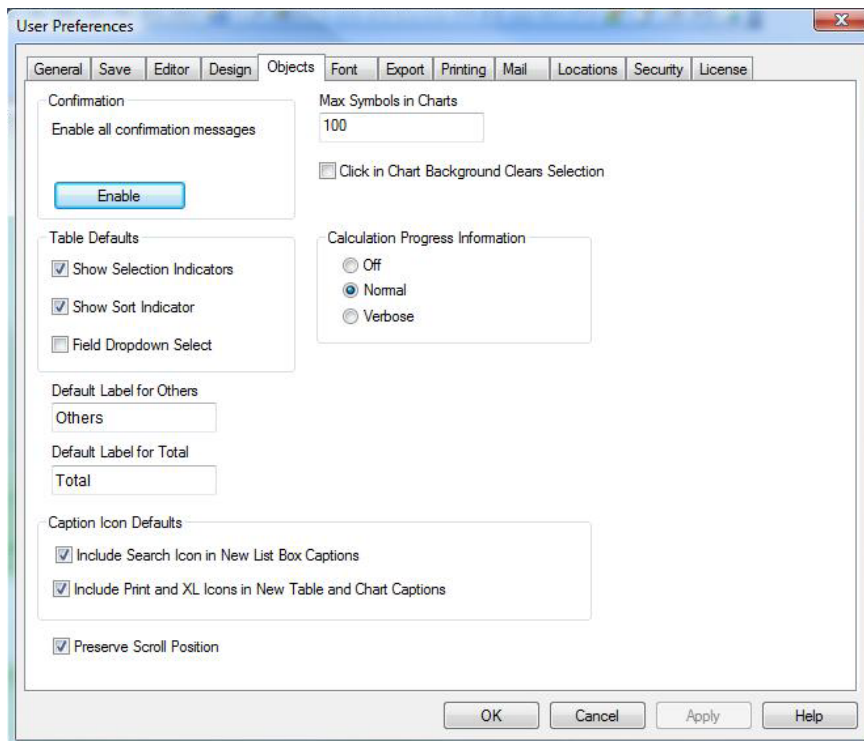
**Always Use Logfiles for New Documents** A log file (.log) will be generated when the script is executed. The file will contain a timestamp for start and finish, the executed script lines, number of rows generated by the script lines and, if the script execution was unsuccessful, an error message.

**File Wizard Sample Size** Specifies the number of records read into the table file wizard in order to analyze a table being read.

**Default Scripting Engine** Choose between **VBScript** and **JScript**.

<b>Design Grid Settings</b>	<p><b>Snap Step (mm)</b> Sets the distance between the snap points when the design grid is displayed.</p> <p><b>Line Distance (mm)</b> Sets the distance between the grid lines when the design grid is displayed.</p>
<b>Default Margin Unit</b>	Here it is possible to select whether the default unit for margins in the <b>Print: Layout</b> page should be cm or inch.
<b>Default Styling Mode</b>	Choose one of the available modes for the object style of all the sheet objects. The mode chosen will be used as default for all new documents.
<b>Default Sheet Object Style</b>	Choose one of the available styles for the sheet object style in this drop-down. The style selected will be used for all sheet objects in the document.
<b>Default Theme</b>	<p>Here it is possible to select a QlikView theme which will be set as the default theme in newly created 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.</p> <p>At the bottom of the drop-down list there is a <b>Browse...</b> command in case the theme file resides in another location than in the default QlikView theme catalog. If no default theme is specified, new documents will be created without default theme.</p>
<b>Always Show Design Menu Items</b>	<p>If this checkbox is marked all design menu options will be available on the context menu at all times.</p> <p>If unchecked some design menu options will only be available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is turned on.</p>

## 12.5 User Preferences: Objects



*User Preferences, Objects*

Here the user can determine a number of default settings for sheet objects.

**Enable all confirmation messages**

When QlikView is first installed on a computer a number of warning dialogs are enabled. Each of these will prompt for confirmation before performing certain actions, such as deleting sheets and sheet objects or sending e-mail. Every warning dialog includes a checkbox stating: "**Do not show this message again**". By checking the box that particular warning dialog is permanently suspended. To turn all previously disabled warning dialogs back on, click the **Enable** button in this group.

**Table Defaults****Show Selection Indicators**

Select this option if column selection indicator (beacons) should be enabled as default for new table boxes, pivot tables and straight tables.

**Show Sort Indicator**

Select this option if an icon indicating the primary sort column should be default for new table boxes and straight tables.

**Field drop-down Select**

Select this option if a drop-down selection icons should be displayed in field columns for new table boxes, pivot tables and straight tables.

**Default Label for Others**

In a number of chart types it is possible to limit the number of plotted data points by setting a **Max** value. All data points that fall outside this limit are collectively grouped as "Others". It is possible to edit the default label for Others here.

**Default Label for Total**

The totals that may be displayed in bar charts, pivot tables and straight tables is given the default label "Total". It is possible to edit the default label for Total here.

**Caption Icon Defaults**

In the 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 enabled at creation. This is recommended for better usability and especially if the document is to be published for the QlikView AJAX 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 enabled at creation. This is recommended for better usability and especially if the document is to be published for the QlikView AJAX client.

**Preserve Scroll Position**

With this setting enabled, QlikView will try to preserve the vertical scroll position of tables and charts with an x-axis scroll bar when a selection is made in another object. The setting must be enabled on the objects' **Layout** page as well.

**Max Symbols in Charts**

It is possible to specify an upper limit to how many data points that will be displayed with symbols. The default number is 100. This feature is only useful in line charts and combo charts, for expressions with both **Line** and **Symbol** checked.

**Click in Chart background Clears Selection**

If this option 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.

**Calculation Progress Information**

In this group it is possible to determine the level of information to be displayed when the calculation of sheet objects require more than one second to complete.

**Off**

No progress information is displayed.

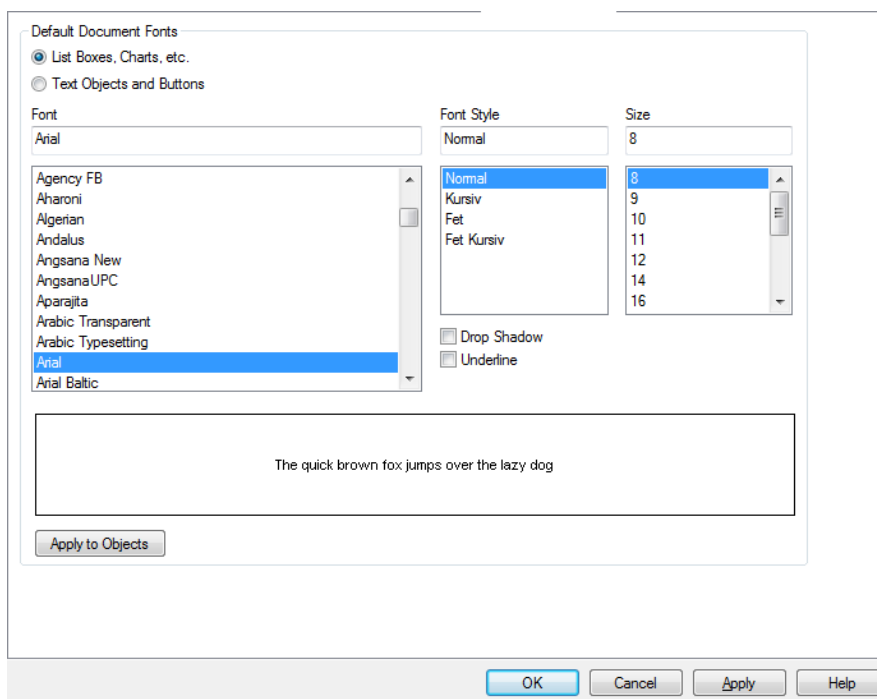
**Normal**

A progress bar is displayed.

**Verbose**

A progress bar and additional text information is displayed.

## 12.6 Font

*The Font dialog*

Here the **Font**, **Font style** and **Size** of the font to be used can be set.

The font can be set for any single object (**Object Properties: Font**), or all objects in a document (**Apply to Objects on Document Properties: Font**).

Further, the default document fonts for new objects can be set on **Document Properties: Font**. There are two default fonts:

1. The first default font (**List Boxes, Charts, etc**) is used for most objects, including list boxes and charts.
2. The second default font (**Text Objects and Buttons**) is used for buttons and text boxes, which are objects that usually need a larger font.



Finally, the default fonts for new documents can be set on **User Preferences: Font**.

For **charts**, **buttons** and **text objects** (except search objects) a font **Color** can also be specified. The color may be **Fixed** (Click the colored button to specify a different color) or it can be dynamically **Calculated** from an expression. The expression must be a valid color representation, which is created using the *Color Functions* (page 375). If the result of the expression is not a valid color representation, the font color will default to black.

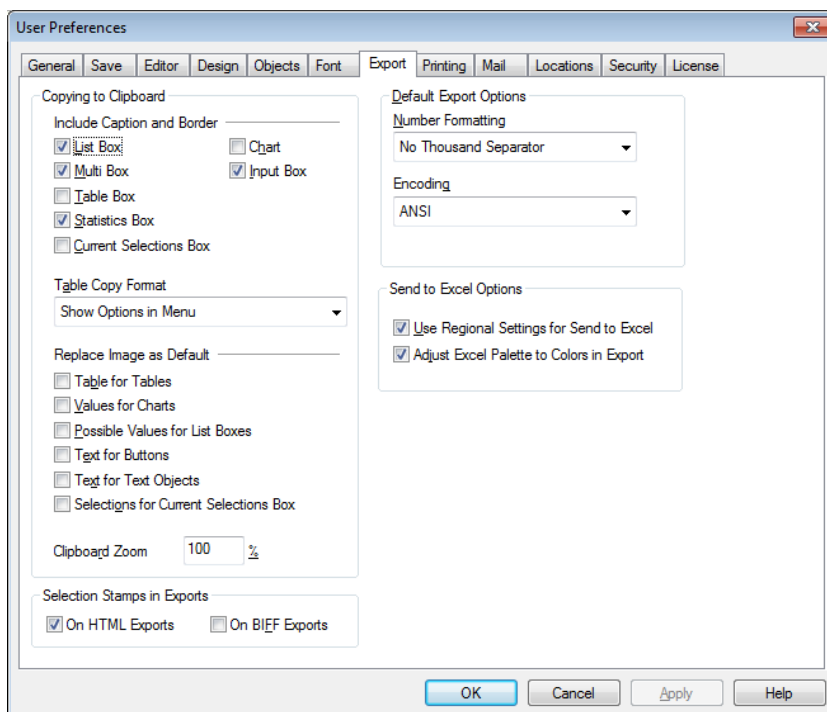
Additional settings are:

**Drop Shadow** If this option is checked a drop shadow will be added to the text.

**Underline** If this option is checked the text will be underlined.

A sample of the selected font is shown in the preview pane.

## 12.7 User Preferences: Export



### *User Preferences, Export*

The **Copying to Clipboard** section covers the preferences for copying sheet objects to the Clipboard.

In the **Include Caption and Border** group it is possible to enter the preferences separately for each of the following sheet object types: List Box, Statistics box, Multi box, Table box, Input box, Current Selections box and Chart for whether to include these layout features or not in the copy process.

It is possible to set a preference for copying tables under **Copy Table Format**. From the command **Copy to Clipboard** in the object menu of any QlikView table it is possible to make additional choices regarding what information should be included at export.

**Full Table** If this option is checked, a formatted table complete with its selection status will be copied. This setting is preferable when copying for presentation purposes.

**Data Area Only** If this option is checked, nothing but the raw data will be copied. This is the preferred setting for quickly moving data between documents etc.

**Show Options in Menu** If this option is checked, both alternatives will always be available as a drop-down list from the **Copy to Clipboard** command.

In the **Replace Image as Default** group it is possible to specify what is placed in **Clipboard** when the **Cut** and **Copy** commands (**Edit** menu) are used. Normally only the bitmap image of a sheet object is copied, but for a number of sheet objects further options are available:

**Table for Tables** Check this option to have tables (table boxes, straight tables and pivot tables) copied in table format rather than as an image.

**Values for Charts** Check this option to have charts copied as underlying table values rather than as an image.

**Possible Values for List Boxes** Check this option to have list boxes copied as possible values rather than as an image.

**Text for Buttons** Check this option to have buttons copied as text rather than as an image.

**Text for Text Objects** Check this option to have text objects copied as text rather than as an image.

**Selections for Current Selections Box** Check this option to have current selections boxes copied as text selection stamp rather than as an image.

The **Clipboard Zoom** setting, which is independent of the current zoom setting of the sheet, determines the size of the copied image. Larger pictures give better picture quality at the cost of increasing size.

In the **Selection Stamps in Exports** group it is possible to determine whether selection stamps should be included when exporting to the file types specified.

**On HTML Exports** Check to include selection stamps when exporting to HTML files.

**On BIFF Exports** Check to include selection stamps when exporting to BIFF (Excel) files.

In the **Default Export Options** group it is possible to set defaults for export formatting.

**Number Formatting** The number formatting of numeric data in QlikView may not always be compatible with other programs due to user defined settings etc. The drop-down offers three choices in **Number Formatting** of numeric data that is to be exported.

**Full Formatting**

Exports 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**).

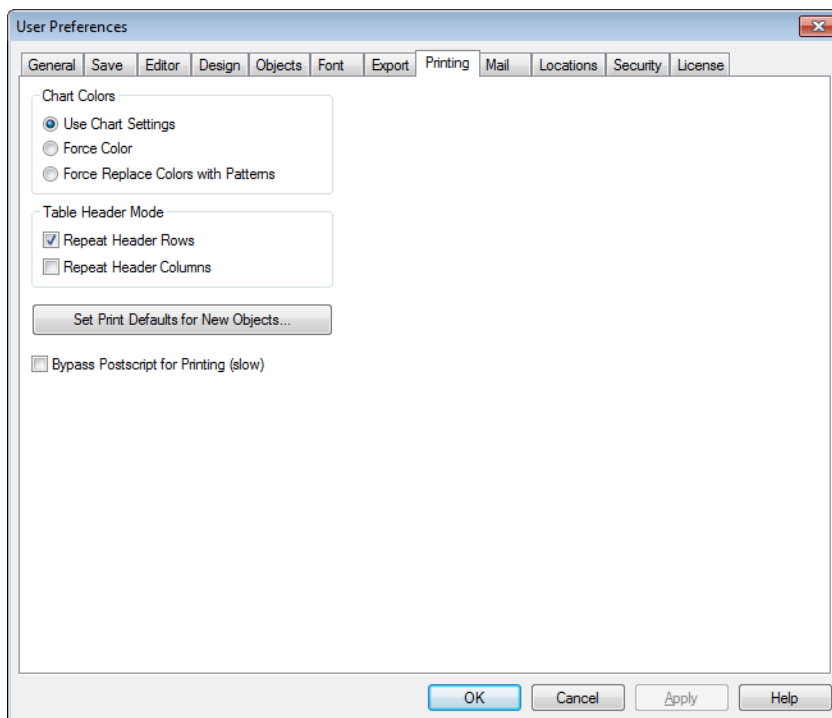
**Encoding** It is possible to set the default character set for export in new documents. Choose one of these option: ANSI, Unicode or UTF-8.

In the **Send to Excel options** group it is possible to set defaults for formatting used by the **Send to Excel** menu command.

**Use Regional Settings for Send to Excel** With this option the regional settings of the operating system are used for the decimal separator when the **Send to Excel** command is used for exporting data. When the alternative is checked, Microsoft Excel may not be able to interpret numerical data correctly. By deselecting this option, a decimal point will be used, regardless of the regional settings.

**Adjust Excel Palette to Colors in Export** 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.

## 12.8 User Preferences: Printing



### *User Preferences, Printing*

In the **Chart Colors** group, it is possible to specify an override setting for all output from charts. It is possible to select one of three alternative settings:

**Use Chart Settings** The intrinsic chart settings regarding color or b/w will always be used.

**Force Color** All bitmap charts will be printed in color, irrespective of their **Colors** settings in the document.

**Force Replace Colors with Patterns** All bitmap charts will be printed in b/w, irrespective of their **Colors** settings in the document.

When printing tables whose contents span over several pages, it is possible to include header rows or columns on every page. This is decided in the **Table Header Mode** group.

- Repeat Header Rows** If this option is checked, header rows will appear on every page.
- Repeat Header Columns** If this option is checked, header columns (in pivot tables) will appear on every page.
- Set Print Defaults for New Objects** This button opens the **Page Setup** dialog where the page margins and page orientation can be set.
- Bypass Postscript for Printing (slow)** Due to the interaction between Microsoft graphics libraries and certain printer Postscript drivers, the sharpness of printouts from **Print Sheet** may occasionally be less than expected. This can be avoided by checking this alternative. It may however result in significantly longer printing times (up to several minutes).

## 12.9 User Preferences: Mail

### *User Preferences, Mail*

Here the user can make settings for sending e-mail from QlikView. In order to use this functionality access to an SMTP server is required .

In the **Sender** group it is possible to specify the **Name** and e-mail **Address** that will appear as sender on e-mail sent by QlikView.

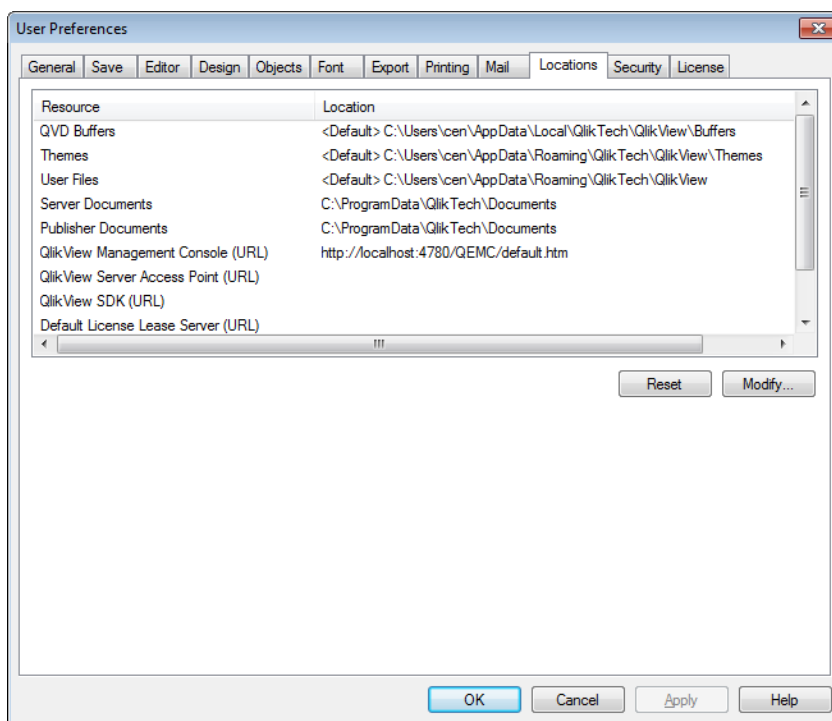
Under **Encoding** it is possible to change the character code page used for sending e-mail, if there are problems with the current setting.

Checking **Send MIME encoded** in order to encode e-mail can also be helpful.

In the **Server** group it is possible to specify settings for the SMTP server to be used for outgoing e-mail from QlikView.

<b>Address</b>	The address (URL or IP number) used for the SMTP server.
<b>Port</b>	The port used by the SMTP server.
<b>Authentication method</b>	Select an authentication method from the list, if authentication is required by the SMTP server.
<b>User ID</b>	The user ID used for authentication.
<b>Password</b>	The password used for authentication.

## 12.10 User Preferences: Locations



### *User Preferences, Locations*

Here the user can determine 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, as well as to specify URLs to the control panels of QlikView Server, QlikView Publisher and QlikView AccessPoint. The tab contains a list of resource locations that can be modified.

The pane at the top of the dialog contains a list of folder locations that can be modified:

**Resource**

The following resource locations can be modified:

**QVD Buffers**

The default storage location of *QVD Files (page 409)* generated via the **buffered** prefix to **load** and **select** statements in the script.

**Themes**

The default storage location for user defined layout themes, see *Theme Maker Wizard (page 603)*.

**User Files**

The default root folder location where the folders containing user bookmarks, user reports and user alerts are stored. Note that by changing this location without moving the folder contents the existing user bookmarks, user reports and user alerts will be lost.

**Server Documents**

Here it is possible to specify the location of the QlikView Server document folder, when applicable.

**Publisher Documents**

Here it is possible to specify the location of the QlikView Publisher source document folder, when applicable.

**QlikView Management Console (URL)**

Here it is possible to specify a URL pointing at the QlikView Management Console, QMC, or QlikView Enterprise Management Console, QEMC, when applicable.

**QlikView Server AccessPoint (URL)**

Here it is possible to specify a URL pointing at the QlikView AccessPoint, when applicable.

**QlikView SDK (URL)**

Here it is possible to specify a URL pointing at the QlikView SDK, when applicable.

**Default License Lease Server (URL)**

Here it is possible to specify a URL pointing to the QlikView License Lease Server, when applicable.

**Publisher Authorization Table (URL)**

Here it is possible to specify a URL pointing to the section access authorization tables created in QlikView Publisher. *Section Access Management* is configured in the QlikView Management Console (QMC). For more information, see the QMC help.

**Location**

The path to the respective folder location.

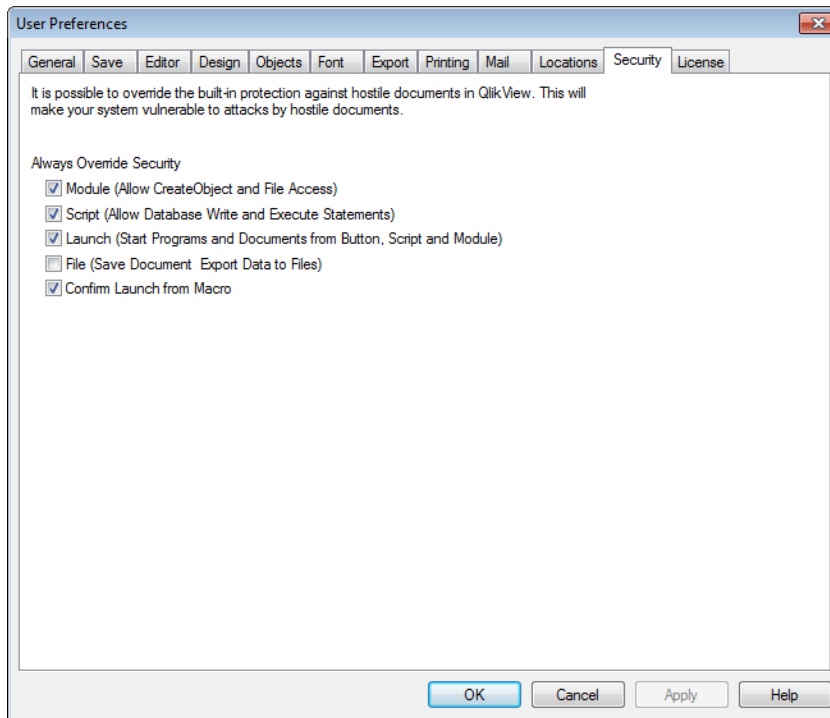
**Reset**

This button resets the location of the selected folder 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 it is possible to browse to the location that is preferred for the selected folder. When a URL resource is modified, this button opens a dialog where it is possible to enter a URL.

## 12.11 User Preferences: Security



*User Preferences, Security*

In this tab it is possible to choose to override one or more parts of the QlikView security measures against hostile macros and scripts embedded in QlikView document. No warning dialogs prompting the user to accept potentially harmful code will be shown. Use these options with caution and only when working with well-known documents.

**Module (Allow CreateObject and File Access)**

Mark this box to disable QlikView checks for macros containing **CreateObject** calls or accessing external files.

**Script (Allow database write and execute statements)**

Mark this box to disable 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)**

Mark this box to disable QlikView checks for starting external programs from QlikView script, module or buttons.

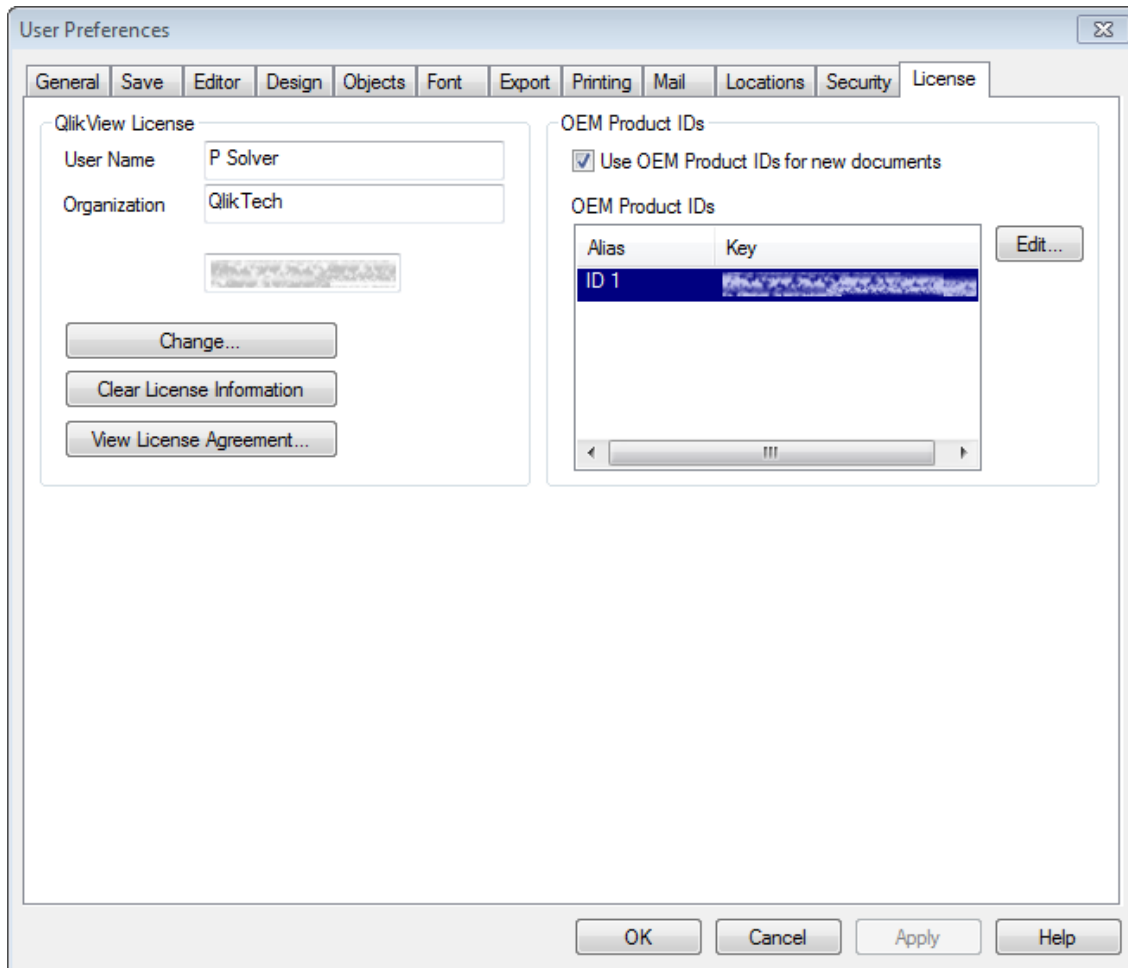
**File (Save doc and export data to file)**

Mark this box to disable QlikView checks for saving or exporting to files with suspect file extensions.

**Confirm Launch from Macro**

Unless this alternative is de-selected, the user will be prompted to confirm the launch of other applications from a macro.

## 12.12 User Preferences: License



*User Preferences, License*

### QlikView License

This dialog shows the present registration information of the QlikView installation.

#### **Change...**

To enter a new serial number and a new control number for the license click this button. Changes take effect only after restarting QlikView.

#### **Clear License Information**

If this button is clicked, the license number will be erased the next time the application is started.

#### **View License Agreement**

Clicking this button will show the license agreement.

### OEM Product IDs

This option is only available with an OEM Partner License.

**Use OEM Product IDs for new documents** Enable this option if the question to use OEM Product ID should be asked for new documents.



**OEM Product IDs** Lists all available OEM Product IDs.

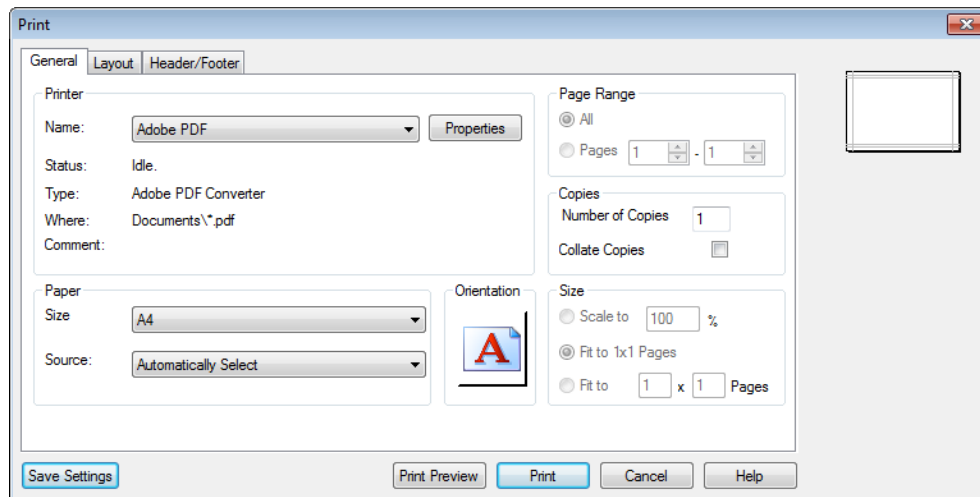
**Edit...** Click this button to edit the Alias for the OEM Product ID.



# 13 Exporting and Printing

## 13.1 Print: General

In this property page you may specify settings relating to the printer and paper. Additional printout settings can be made in the *Print: Layout (page 92)* and *Print: Header/Footer (page 93)* property pages.



*Print, General*

In the **Printer** group available printers are listed in a drop-down list. You can access the printer **Properties** by clicking this button.

In the **Paper** group selections of paper **Size** and **Source** (tray) are made.

Other options in this property page allow you to change the paper **Orientation**, specify the **Page Range** to be printed and specify the **Number of Copies** and if you want to **Collate Copies**.

The **Size** group offers three different scaling options:

- |                           |   |
|---------------------------|---|
| <b>Scale to _ %</b>       | Check this option and enter a percentage number to increase or decrease the scale of the printout.                                  |
| <b>Fit to 1x1 Pages</b>   | Check this option to have the printout scaled to the paper size. You may get a better result if you change the <b>Orientation</b> . |
| <b>Fit to _ x _ Pages</b> | Check this option to have the printout scaled to the specified number of pages.   |

---

### Note!

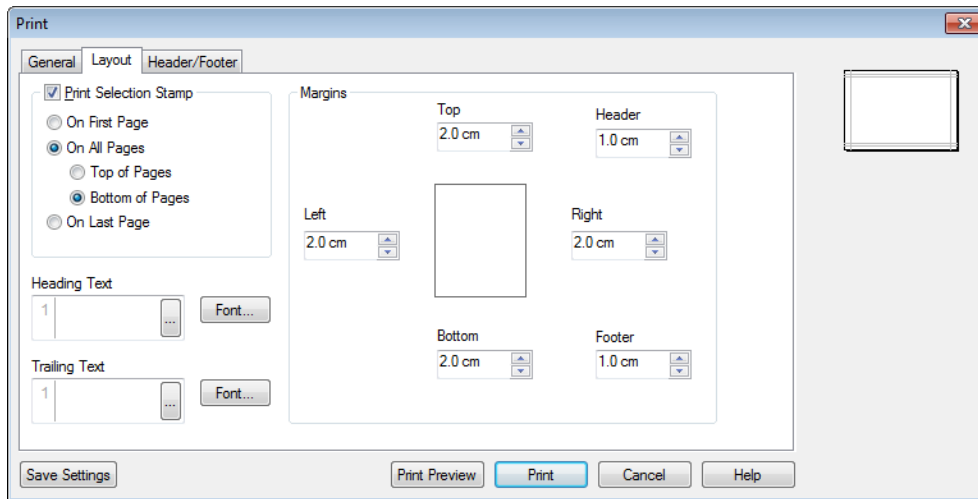
If you arrived at the **Print** dialog by using the command **File: Print Sheet**, the **Size** group will not be available, but be replaced by the **Sheet options** group where you can determine if you want to print just **This Sheet** or **All Sheets**, and if you want the printout to **Draw Background** (wallpaper).

---

The following buttons are also available:

- |                      |  |
|----------------------|--|
| <b>Save Settings</b> | Click this button to save your settings for header and footer while you continue working with this dialog.     |
| <b>Print Preview</b> | This button opens a window showing a detailed <i>Print Preview (page 96)</i> of the current, printable object. |
| <b>Print</b>         | Click this button to effect the print command and close this dialog.   |
-

## 13.2 Print: Layout



*Print, Layout*

In this tab settings for **Print Selection Stamp** and **Margins** can be made. Additional printout settings can be made in the *Print: General* (page 91) and *Print: Header/Footer* (page 93) property pages.

**Print Selection Stamp** In the **Print Selection Stamp** group, the option to include the relevant current selections (i.e. current selections that affect the current object) in the printout is possible. The text "Selection status" will be displayed in the printout, followed by a list of fields and field values. The following options determine on what pages the **Current Selections** will be included: **On First Page**, **On All Pages - Top of Pages**, **On All Pages - Bottom of Pages** and **On Last Page**.

**Heading Text** In the edit box specify a text to be printed before the printed sheet object. This text may be a *Calculated Formula* (page 871). By clicking on the ... button the **Edit Expression** dialog opens for easier editing of long formulas. The **Font** button next to the edit box allows the selection of a separate font for the text.

**Trailing Text** In the edit box specify a text to be printed after the printed sheet object. This text may be a *Calculated Formula* (page 871). By clicking on the ... button the **Edit Expression** dialog opens for easier editing of long formulas. The **Font** button next to the edit box allows the selection of a separate font for the text.

**Margins**

In the **Margins** group you can define the margins around the printed object. Changes can be seen in the preview pane at the right of the **Print** dialog. Measurements are in mm, cm or ". The default unit is set in the **User Preferences: Design** page.

**Top**

Specifies the distance between the top of the paper and the upper border of the printed object.

**Header**

The **Header** value is the distance between the header text and the top of the paper. 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**

The **Footer** value is the distance between the footer text and the bottom of the paper. For the footer text to appear this value must be smaller than **Bottom**.

The following buttons are also available:

**Save Settings**

Click this button to save your layout settings while you continue working with this dialog.

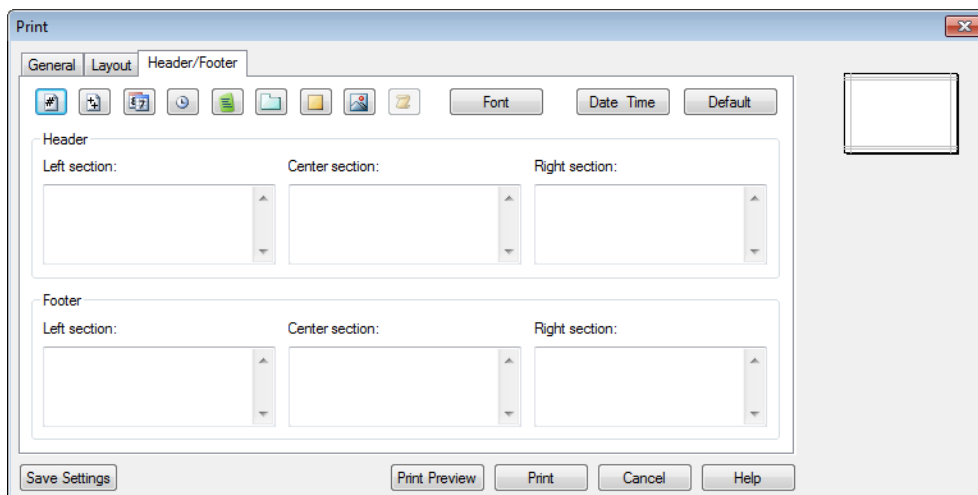
**Print Preview**

This button opens a window showing a detailed *Print Preview* (page 96) of the current, printable object.

**Print**

Click this button to effect the print command and close this dialog.

## 13.3 Print: Header/Footer



*Print, Header/Footer*

In this property page you may specify settings for **Header** and **Footer**. Additional printout settings can be made in the *Print: General (page 91)* and *Print: Layout (page 92)* property pages.

The following buttons are used for inserting control codes for specific system information into any of the text panes. Control codes may also be typed directly:

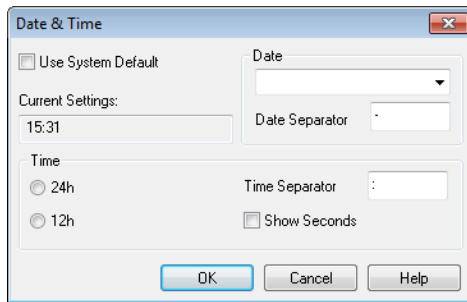
<b>Page</b>	Click this button or enter the code <code>&amp;[Page]</code> to insert the page number.
<b>Pages</b>	Click this button or enter the code <code>&amp;[Pages]</code> to insert the total number of pages. The use of this option in reports with banding may cause long delays for calculation before printing can commence. You will therefore be warned when this situation occurs.
<b>Date</b>	Click this button or enter the code <code>&amp;[Date]</code> to insert the current date. Date format can be set in the <i>Date &amp; Time (page 95)</i> dialog.
<b>Time</b>	Click this button or enter the code <code>&amp;[Time]</code> to insert the current time. Time format can be set in the <b>Date &amp; Time</b> dialog.
<b>File</b>	Click this button or enter the code <code>&amp;[File]</code> to insert the filename.
<b>Sheet</b>	Click this button or enter the code <code>&amp;[Sheet]</code> to insert the name of the sheet. This option is not available when printing reports.
<b>Title</b>	Click this button or enter the code <code>&amp;[Title]</code> to insert the caption title of the printed object. This option is not available when printing reports.
<b>Picture</b>	Click this button to import a picture from the <b>Select Image</b> dialog. The picture will be printed as graphics in the header or footer pane. You may also manually enter the code <code>&amp;[Picture=filename]</code> where <i>filename</i> is the complete filename and path of the file containing the graphics.
<b>Report</b>	Pressing this button or entering the code <code>&amp;[Report]</code> causes the title of the report to be printed. This option is only available when printing reports.

The **Header** and **Footer** groups allow the above settings to three panes: **Left section**, **Center section** and **Right section**. Simply click in the desired pane to position the cursor and then click a button or type in the code.

The following buttons are also available:

<b>Font</b>	This button opens the <i>Font (page 491)</i> dialog.
<b>Date &amp; Time</b>	This button opens the <i>Date &amp; Time (page 95)</i> dialog.
<b>Default</b>	Click this button to reset the default settings for header and footer.
<b>Save Settings</b>	Click this button to save your settings for header and footer while you continue working with this dialog.
<b>Print Preview</b>	This button opens a window showing a detailed <i>Print Preview (page 96)</i> of the current, printable object.
<b>Print</b>	Click this button to effect the print command and close this dialog.

## 13.4 Date & Time



*The Date & Time dialog*

Here the preferred display of date and time can be set.

<b>Use System Default</b>	Enable this option to apply the system (e.g. Windows) time and date formats.
<b>Current Settings</b>	A view of the current date and time format.
<b>Date</b>	The date format can be set here. Select format from the drop down list.
<b>Date Separator</b>	Choose the character to be used as date separator.
<b>Time</b>	In the Time group settings for the time format can be set.
<b>24h</b>	Enable this option to show time in the 24 hour notation.
<b>12h</b>	Enable this option to show time in the 12 hour notation.
<b>Time Separator</b>	Choose the character to be used as time separator.
<b>Show Seconds</b>	Enable this option to show seconds in the time format.

## 13.5 Print Sheet

To open this dialog, choose **Print Sheet** from the **File** menu. This dialog 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:

<b>This Sheet</b>	By selecting this option only the current sheet will be printed.
<b>All Sheets</b>	By selecting this option all sheets in the document will be printed.
<b>Draw Background</b>	Mark this check box to include the sheet background (wallpaper) when printing.

## 13.6 Print Preview

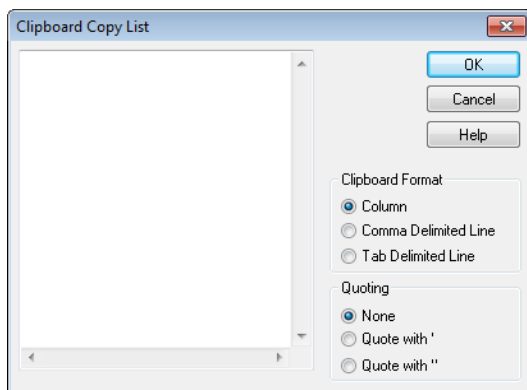
Use the preview feature when you want a detailed print preview to see how a printable sheet object will appear in the printout. The magnifying glass toggles the display between the preview size where you see the whole page and the actual 100% size.

<b>Print</b>	Transfers control to the <i>Print: General (page 91)</i> dialog, from where you can print the active object.
<b>#pages drop-down</b>	This drop-down menu offers a quick way to change which pages to display in the preview.
<b>Previous</b>	If the preview contains several pages, you can click this button to view the previously shown page.
<b>Next</b>	If the preview contains several pages, you can click this button to view the next page.
<b>Add Page</b>	Extends the preview to include another page if the active object does not fit in one page.
<b>Remove Page</b>	Removes the current preview page.
<b>Close</b>	Closes this dialog.
<b>Help</b>	Opens QlikView help.

## 13.7 Copy Mode

Switches from logic mode to copy mode. In copy mode the clicked values are copied to the **Clipboard** without changing the logical state of the running QlikView document. When copy mode is on, the *Clipboard Copy List (page 96)* dialog opens. In this dialog you build a list of values to be copied.

### Clipboard Copy List



*The Clipboard Copy List dialog*

Opens from *Copy Mode (page 96)* in the **Edit** menu. This dialog simplifies copying to the clipboard. As long as it is open, QlikView will be in copy mode and values clicked will automatically be copied to the **Clipboard Copy List**. The QlikView logic is disabled while the document is in copy mode.

<b>Clipboard Format</b>	In this group, the format of the copy list is set. <b>Column</b> , <b>Comma Delimited Line</b> and <b>Tab Delimited Line</b> are the available options.
-------------------------	---



<b>Quoting</b>	<p>In this group, the quoting for the selected elements can be set.</p> <p><b>Quote with ' </b>will enclose all elements with single quotes. This is useful if the elements copied are to be pasted into the script as field values.</p> <p><b>Quote with " </b>will enclose 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.</p> <p><b>None</b> will leave the elements unquoted.</p>
<b>OK</b>	Closes the <b>Clipboard Copy List</b> dialog and transfers its content to the Windows <b>Clipboard</b> .
<b>Cancel</b>	Closes the <b>Clipboard Copy List</b> dialog without transferring its content to the Windows <b>Clipboard</b> .

## 13.8 Export/Export Contents...

Opens the **Save As** dialog. Here a name, path and file type for the exported data can be specified.

The file can be saved as any of the following formats: Comma Delimited, Semicolon Delimited, Tab Delimited, Hypertext (HTML), XML and Excel (xls). The default format is \*.qvo (QlikViewOutput), a tab separated file.

---

**Note!**

When exporting to a qvo file, the data is exported as it appears in the QlikView document. When exporting to Excel, the underlying data is exported, but displayed formatted in Excel.

---



# 14 Logic and Selections

Making selections is the main interaction method with QlikView. Selections filter out a subset of the data that has been loaded into QlikView.

You use selections to point out something you want to know more about, and QlikView responds by color-coding possible values and excluded values in different colors.

## 14.1 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, selected. The selection may affect the states of a large number of values in the other sheet objects, on the current sheet as well as on other sheets.

White cells represent optional field values, and gray cells represent field values excluded by selections.

When a selection is made, natural joins are dynamically evaluates between all the associated tables.

Selections can be made in most kinds of sheet objects.

To deselect a previously made selection, click it again. Alternatively, 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.

## 14.2 Color Codes

The state of a field value is shown by the color of its cell.

The following color scheme is used by default:

- selected green
- optional white
- excluded gray

There are also other color codes that can be used under special circumstances:

- alternative white/yellow
- locked blue
- red as a result of forced exclusion

The default colors of the default (Classic) color scheme can be modified by selecting another color schemes in the **Selection Appearance** group on the **Document Properties: General** page.

When the *Show Alternatives* (page 478) option is enabled, QlikView displays non-selected cells in the corresponding list box as alternative values (white or light yellow), unless they are excluded by selections in other fields. They are however logically excluded, and not included in calculations made on possible (optional and selected) values. When the **Show Alternatives** option is disabled, QlikView displays non-selected cells as excluded (gray).

The selection style **Windows Checkboxes** does not make use of this color scheme.

## 14.3 Selection Styles

QlikView supports a number of different ways of presenting field values and making selections in list boxes, charts and other objects. The **QlikView Classic**, **Corner Tag** and **LED** styles all use color coding for indication of selected, possible and excluded values. The **Windows Checkboxes** style mimics the standard Windows interface with a check box at each value. The **LED Checkbox** style uses the color-coding in combination with the **Windows Checkboxes** style.

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.

Which style to use can be controlled in a few different ways:



- on the **Document Properties: General** page ( *Document Properties: General (page 424)* )
- on the **User Preferences: General** page ( *User Preferences (page 69)* )
- on the **List Box Properties: Presentation** page ( *List Box Properties: Presentation (page 487)* ).

## Representation of Logical State

The logical state of a field value 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-selections and forced exclusion (sometimes referred to as NOT selection).

With the **Windows Checkboxes** 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 state	QlikView color-coded styles	Windows check box styles (icons)
Possible	White/black	<input type="checkbox"/>
Excluded	Gray/light gray	<input type="checkbox"/>
Selected	Green/black	<input checked="" type="checkbox"/>
Selected excluded	Gray/light gray	<input checked="" type="checkbox"/>
Locked	Blue/yellow	
Locked excluded	Gray/light gray	
AND-selected	Green/black with &	<input checked="" type="checkbox"/> &
Subject to forced exclusion	Red/black with !	<input checked="" type="checkbox"/> !

## Logical Behavior

The logical behavior of the **Windows Check Boxes** and the **LED Checkbox** style differs in two ways from that of the color-coded styles.

1. All clicks under the **Windows** and **LED Check Boxes** style are considered toggle selections, i.e. they work as if the Ctrl key is held down while clicking under the color-coded styles.

- All list boxes are treated as if the **Show Alternatives** option is selected, regardless of the actual setting of this property.

## 14.4 Indicators

Indicators (or beacons) are colored dots that sometimes appear in tables, on *tabs* and in the right-hand corner of the *Status Bar* (page 67). Indicators are there simply to remind you of selections that have been made in fields that are not available on the sheet that you are currently viewing. Since all sheets of a QlikView document are fully interconnected at all times, such selections will most likely affect what is displayed on the active sheet - even when they are not immediately apparent! This is the main reason for having indicators.

Selection indicators may also appear in the upper right-hand corner of the data fields in the QlikView tables: table boxes, pivot tables and straight tables. This is a useful option as the selections in tables are not themselves color-coded. The option is selected/deselected in the *User Preferences: Objects* (page 78) dialog.

Selection indicators will appear in the current selections box as well as in the free-floating current selections window, in order to distinguish between selected and locked values.

### Indicator Color Scheme

The color of the indicators follows the general color scheme:

<b>Green</b>	Selected values
<b>Blue</b>	Locked values
<b>Red</b>	Values in AND-mode that are included in a forced exclusion

#### Example:

The picture below shows what the different indicators look like in the current selections box.



## 14.5 Multiple Selection Within a Field

Multiple selections within a field, i.e. allowing several values from a field, can be made in a number of ways:

- Position the cursor in the list box and 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 and the press Enter. 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 multiple selections 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 canceled, the previous selections will, by default, be recovered.

Multiple selections are normally interpreted as a logical OR, which means that QlikView will find a solution that is associated to either one of the selected field values.

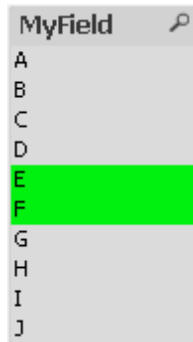
## 14.6 Moving Selections

You can use the keyboard keys to move the current selection in an active list box or in an opened multi box field.

- i** Moves the current selection(s) one step downwards in the list box. When the last value is reached, the selection will wrap back to the first value. When no selection is made, the list box will just scroll down one cell at a time.
- h** Moves the current selection(s) one step upwards in the list box. When the first value is reached, the selection will wrap back to the last value. When no selection is made, the list box will just scroll up one cell at a time.
- Page Up** Moves the current selection one interval equal to the distance between the outermost selections, upwards in the list box. When the first value is reached, the selections will wrap back to the last set of values. When no selection is made, the list box will just scroll up one page at a time.
- Page Down** Moves the current selections one interval equal to the distance between the outermost selections, downwards in the list box. When the last set of values value is reached, the selections will wrap back to the first set of values. When no selection is made, the list box will just scroll down one page at a time.
- Home** Moves the current selection(s) to the top of the list.
- End** Moves the current selection(s) to the end of the list.

### Example:

In the list box below, with E and F selected, using the different keys produces the following result:



- i** F and G will be selected.
- h** D and E will be selected.
- Page Up** C and D will be selected.
- Page Down** G and H will be selected.
- Home** A and B will be selected.
- End** I and J will be selected.

## 14.7 Locking Selected Field Values

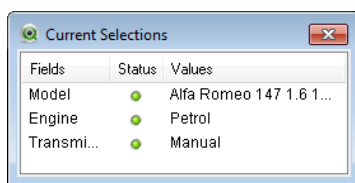
When selecting an excluded field value, previous selections in conflict with the new choice are canceled. To prevent a previous selection from getting canceled this way, it is possible to lock all the selections in a list box by opening its **Object** menu and choosing the **Lock** command.

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 trying 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.

## 14.8 Current Selections



In the **Current Selections** window, selections are listed by field name and field value. This window remains on top of every sheet and helps keeping track of the selection status in the document. In the **Status** column *Indicators (page 101)* are used for distinguishing between selected and locked values. It is opened from the **View** menu or the **Current Selections** button in the toolbar.

When you right-click in the **Current Selections** window without having highlighted a field, the float menu that contains the following commands is displayed:

**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 working with a document. It will be left open when you close the current QlikView document, and its contents will change if you open another document. Just as the sheet objects, it is dynamically updated as soon as new selections are made, which gives you an overview of the current selections at any time.

The maximum number of distinct selected values to be shown in the **Current Selections** windows is specified in the **User Preferences** dialog ( *User Preferences (page 69)*). 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.

## 14.9 Selections in Other Objects

You can make selections directly in fields in most objects. In tables, the color of the cell becomes green to mark its value as selected. Click to select a value or paint with the mouse to select multiple values.

You can also make selections by searching.

The following section describes how you can make selections in the different types of objects. If you find that you cannot make selections in an object as described below, the chart may be in **Detached** mode or **Read Only** mode.

## Statistics Boxes

In statistics boxes you can click on some of the statistical quantities such as **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. You can make selections and searches in this list just as in a list box.

## Table Boxes

You can make selections in table boxes 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. You can make selections and searches in this list just as in a list box.

## Slider/Calendar Objects

In sliders, where a single field is the basis, you can 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. You can select a date or a period of time 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

In these objects, you can make selections in different ways:

- By clicking on a single data point or painting over several data points inside the plot area. 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).
- By clicking or painting in the chart legend (except when the legend is indicating chart expressions rather than dimension values).
- By clicking or painting over the dimension axes and its labels (except scatter charts). The corresponding field values will be selected.
- 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 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.

## Pie Charts

You can make selections 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).

You can also make selections by clicking or painting in the chart legend.

## Block Charts

You can select individual blocks in block charts. The drilldown functionality is used to refer to a certain dimension - selecting the first block refers to the first dimension, selecting a second block within the first refers to the second dimension etc.

You can also select several blocks by painting an area. This selected area is marked green until you release the mouse button. 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 since there is no dimension defined.

## Straight Tables

You can make selections in the dimension columns of a straight table by clicking on a cell or painting over several cells. The selected area is marked green until you release the mouse button.

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. You can make selections and searches in this list.

You can also make selections 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

You can make selections in a pivot table in the dimension columns/rows by clicking on a single cell. The selected cell is marked green until you release the mouse button.

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. You can make selections and searches in this list.

You can also make selections 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.

## 14.10 Search

As an alternative to making selections by clicking values in QlikView objects, you can make selections through text or numeric search.

### Making Selections Using Search

The following section describes how to make selections using text search, and how QlikView responds to different commands. Perform the following steps:

1. Click on the list box caption, and type the search string. The string is not case sensitive.

The search string appears 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.

2. Press Enter or click one of the cells in the result to select that value. 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 press Enter or Esc, or when you click in the layout. You can also close the search box by clicking the x-icon in the search box. The search box is sizable and will retain its size when it is opened again.

## Search in Multiple List Boxes

All active list boxes will be included in the search. To search in multiple list boxes, Shift-click on their captions to make them active. It will not be possible to press Enter to select the resulting values as long as there are optional values in several of the active list boxes.

If a selection has already been made, there are two ways in which a search can be interpreted:

- Search only among the optional values
- Search among all values, i.e. including the excluded values in the search

To set the search mode to work in, it is possible to 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.

---

**Note:**

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.

In normal search (no wildcards are used), QlikView will look for words that begin in the same way as the search string. If the search string contains several words separated by blanks, QlikView will interpret it as several search strings and display field values that contain either of the strings.

If wildcards are used, only those records that match the entire search string will be displayed, that is a blank does not imply a logical OR. Wildcards may appear several times in the search string, regardless of their location. The following wildcards can be used:

- \* zero or more characters
- ? any single character
- ^ whole string

Press Enter to select the values found, or Esc to cancel the operation.

### Example:

The following results are returned depending on the selected search method:

String	Search method	Result
ab	Normal	Returns all values that have words that begin with ab.
a*	Wildcard	Returns all values that begin with the letter a.
*b*	Wildcard	Returns all values that contain the letter b.

## Fuzzy Search

Fuzzy search is similar to standard search, with the exception that it compares and sorts all field values according to their degree of resemblance to the search string. Fuzzy search is especially useful in situations where misspelling is an issue. It can also help you find multiple values that are nearly identical to each other.

When a fuzzy search is made, a tilde-character (~) is displayed in front of the search string.

If starting a text search with a tilde-character, the text search window will open 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 press Enter, the first value in the list will be selected.

## Numeric Search

You can make selections using 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 following relational operators:

Operator	Description
>	greater than
>=	greater than or equal
<	less than
<=	less than or equal

### 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

## Initial Search Mode

When starting to type text, the behavior may differ: in some cases, QlikView adds wildcards to the search string, to facilitate a wildcard search.

The preferred search mode can be set in the object properties and in **User Preferences**.

## Search String Evaluation

After a search string has been entered or edited, QlikView evaluates which one of the above described search behaviors to select.

If the search string contains wildcard characters, a wildcard search will be made instead of normal search.

It is always possible to change the search mode simply by deleting or adding wild cards, a greater than (>) or smaller than (<) symbol in the search string.

## Associated Search

The search box contains a chevron (>>) to the right. If you click it, the search box is expanded to the right and a secondary result set is displayed next to the primary result set. This secondary list contains search matches in other fields. It is now possible to click in the secondary result set and make temporary selections. Such selections will narrow down the result in the primary result set. Once you make a selection in the secondary list, you can enter a new search string before making the selection in the primary list. Finally, when you make a selection in the primary result set, the secondary result list is closed.

## Advanced Search

For complex search expressions it is possible to use the **Advanced Search** dialog, 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 can type any valid QlikView layout expression (*Edit Expression Dialog (page 757)*). 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.

### Example:

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

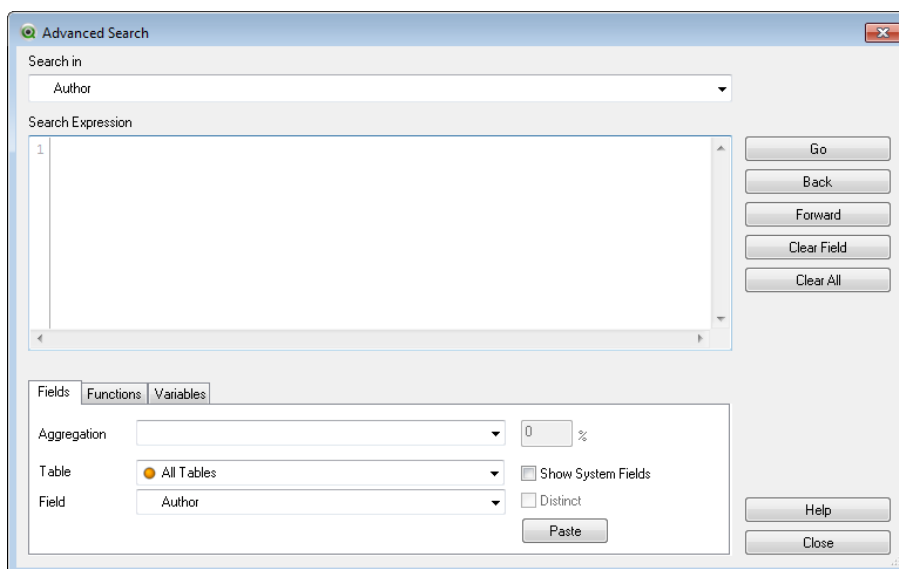
if e.g. invoked from a list box containing the field MyField, the search would return all field values either starting with the letter A or ending with the letter Z.

### Example:

=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.

## 14.11 Advanced Search Dialog



*The Advanced Search dialog*

The **Advanced Search** dialog provides a convenient way of formulating complex search queries to be applied on QlikView fields. Unlike the standard *Search (page 105)* box that appears when starting a search

by typing when a list box is active, the result of the search is not interactively visible in the layout until submitting the search by pressing the **Go** button. The dialog can be kept open while working in the QlikView layout. The dialog is fully sizeable in order to facilitate editing of large and complex expressions.

<b>Search in</b>	The field in which the search will be made. When entering the dialog it will be set to the field of the active list box. It is possible to change search fields by using the drop-down at any time.
<b>Search Expression</b>	This is where the search expression is typed. The same rules apply as when using the normal search box.
<b>Go</b>	Applies the search to the search field
<b>Back</b>	QlikView remembers the last 100 selections. Click this button to go one step back in the list of selections.
<b>Forward</b>	Click this button to go one step forward in the list of selections (equivalent to canceling the last <b>Back</b> command). This only possible if the <b>Back</b> command has been used immediately before.
<b>Clear Field</b>	Clears selections in the current search field.
<b>Clear All</b>	Clears all selections in the document
<b>Help</b>	Opens the <b>Help</b> dialog for advanced search.
<b>Close</b>	Closes the dialog.

The the pane with three tabs at the bottom of the dialog can help you when you build advanced search expressions.

## Fields

Select the **Fields** tab to access controls for pasting syntax relating to QlikView field data.

<b>Aggregation</b>	In this drop-down it is possible to choose from the statistical aggregation functions available in the QlikView layout.
<b>Table</b>	In this drop-down it is possible to select a specific table from which to pick fields in order to make navigation in the <b>Field</b> drop-down easier.
<b>Field</b>	This drop-down lists all the fields available. The list may be narrowed down by selecting a specific table in the <b>Table</b> drop-down above.
<b>Show System Fields</b>	If this check box is marked, the list containing the fields of the document includes the system fields.
<b>Distinct</b>	The statistical functions are by default calculated on the number of occurrences in the original table. Sometimes, however, duplicates are not to be calculated. If this is the case, mark this check box before pasting the function.
<b>Paste</b>	Pastes the selected function or just the field into the <b>Search Expression</b> edit box. A percentage can be given when using the fractile function.

## Functions

Select the **Functions** tab to access the controls for pasting syntax relating to QlikView general functions.

<b>Function Category</b>	In the drop-down it is possible to select a category of functions in order to make navigation in the <b>Function Name</b> drop-down easier.
--------------------------	---

**Function name** In the drop-down it is possible to 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 dialog there is a pane showing the argument syntax of the function selected in the **Function Name** drop-down.

## Variables

Select the **Variables** tab to access the controls for pasting syntax relating to QlikView variables.

**Variables** In the drop-down all currently defined variables in the document are found.

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

## 14.12 AND-Mode in List Boxes

Multiple selections within a field are by default interpreted as logical OR, which means that data associated to any of the selected field values will be shown in the other fields.

It is however possible to set list boxes to AND-mode instead. When multiple selections are made in a list box set to AND-mode, only data associated to all of the selected field values will be shown in the other fields.

### Example:

Your data source contains information about which customers have purchased different articles.

When you select a couple of articles in the default mode, QlikView will show you the customers that have purchased any of the selected articles.

When you select a couple of articles in AND-mode, QlikView will show you the customers that have purchased all of the selected articles.

## AND-Mode Prerequisites

The mode for the field is set in **List Box Properties: General** tab.

When **And mode** is activated, an ampersand ("&") will be displayed in front of selected values.

If clicking and holding a value for a moment, the selection will switch from (**AND**) selected (green) to **NOT** selected (red). The ampersand will also be replaced by an exclamation mark ("!"). The **NOT** selection which is a forced exclusion of the value or values so marked, can only be made when a list box is in **And mode**.

## AND-Mode criteria

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 of no more than two columns,
- the table must not contain any duplicate records, and

- the field must be loaded with the distinct qualifier. If the table is loaded using a select statement, you must use a preceding load distinct \*.

## Setting a List Box in AND-Mode

The following procedure describes how to set a list box in AND-mode instead of the default mode (logical OR). Perform the following steps:

1. Make sure that the AND-mode criteria are fulfilled.
2. Right-click on the list box and select **Properties**.
3. On the **General** tab, mark the **And mode** check box.

## Forced Exclusion

Forced exclusion (sometimes referred to as NOT-selection) is closely related to the AND-selection. The forced exclusion makes it possible to explicitly exclude a field value, which means that the solutions found by QlikView may not be associated with the excluded value.

The forced exclusion is made by clicking a cell and keeping the mouse button pressed 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 forced exclusion can only be made on a field that is in AND-mode.

## 14.13 Circular References

When keys between at least three tables in the data model form a circle, circular references are created. This would generate a loop in QlikView's associative logic. To avoid circular references, QlikView will set one or more tables to be loosely coupled. If this happens, a warning dialog will be displayed when you reload the script.

You are provided with an overview of the loosely coupled tables on the **Tables** tab of the **Document Properties** dialog.

### Example:

Data is loaded from three tables that include the following:

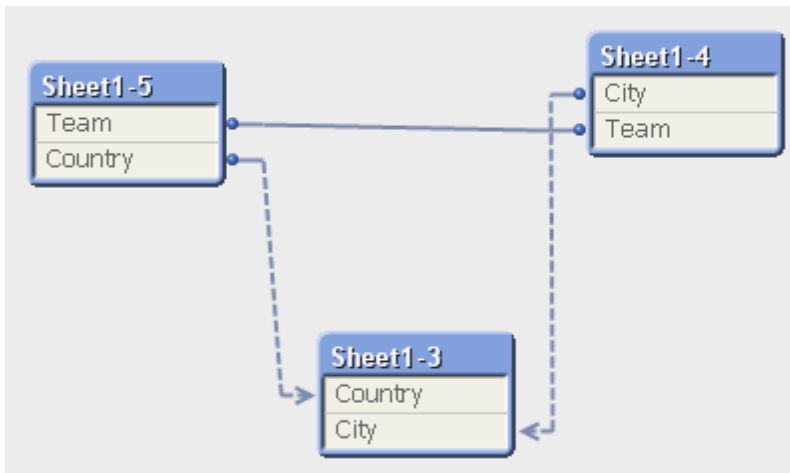
- Names of some national soccer teams
- Soccer clubs in some cities
- Cities of some European countries

The image shows three spreadsheets side-by-side. The first spreadsheet, titled 'NationalTeams', has columns 'Country' and 'Team' with rows: Germany Die Mannschaft, Italy Azzurri, Spain La Roja. The second spreadsheet, titled 'Clubs', has columns 'City' and 'Team' with rows: Barcelona Barcelona, Hamburg Altona, Madrid Real Madrid, Milano Milan, Munich Bayern München, Rome Lazio, Turin Juventus. The third spreadsheet, titled 'Cities', has columns 'Country' and 'City' with rows: Germany Hamburg, Germany Munich, Italy Milano, Italy Rome, Italy Turin, Spain Barcelona, Spain Madrid.

This data structure is not very good, since the field name *Team* is used for two different purposes: national teams and local clubs. The data in the tables creates an impossible logical situation.

When loading the tables into QlikView, QlikView determines which of the data connections that is least important, and loosens this table.

Select **File/Table Viewer** to see how QlikView interprets the relevance of the data connections:



The table with cities and the countries they belong to is now loosely coupled to the table with national teams of different countries and to the table with local clubs of different cities.

## Solving circular references

When circular references occur, you need to edit the QlikView script by assigning a unique name to one of the fields with identical names. Perform the following steps:

1. Open the **Script Editor**.
2. Edit the LOAD statement for one of the duplicate field names. In this example, the LOAD statement of the table that holds the local teams and their cities would include a new name for *Team*, for example *LocalClub*. The updated LOAD statement reads:

```
LOAD City, Team as LocalClub
```

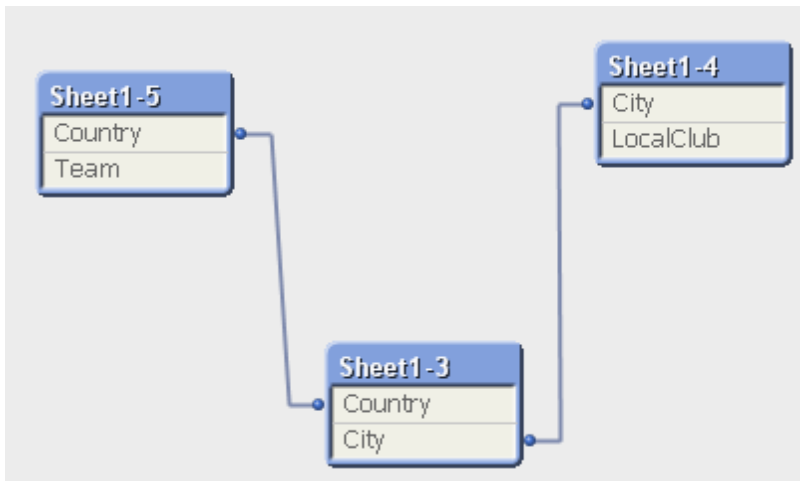
3. Reload the script.

You now have logic that works through out all the tables. In this example, when Italy is selected, the national team, the Italian cities and the local clubs of each city are associated:

Country	Team	City	LocalClub
Germany	Azzurri	Barcelona	Altona
Italy	Die Mannschaft	Hamburg	Barcelona
Spain	La Roja	Madrid	Bayern München
		Milano	Juventus
		Munich	Lazio
		Rome	Milan
		Turin	Real Madrid

When you open the **Table Viewer**, you see that the loosely coupled connections are replaced with regular connections:





## 14.14 Loosely Coupled Tables

When data that includes circular references is loaded into QlikView, loosely coupled tables are created automatically. This prevents that the circular references create a loop in the QlikView internal logic. The loosely coupled tables need to be handled in order to visualize data in a way that is expected and understandable.

### See also:

*Circular References (page 111)*

*Intentionally Creating Loosely Coupled Tables (page 925)*

## 14.15 Alternate States

### Description

You can use **Alternate States** to perform comparative analysis between sets of multiple data items, for example for basket analysis purposes. A state holds a set of selections.

The QlikView developer can create multiple states within a QlikView document and apply these states to specific objects within the document. The end-user can create copies of these objects (server objects) and then put those objects into different states.

The most important difference between **Detach** and **Alternate States** is that ALL objects in a given state will respond to user selections made in that state. Objects in a given state are not affected by user selections in the other states.

**Alternate States** functionality is not accessible in the load script.

#### Note!

Alternate States functionality is enabled by a QlikView developer and should be used with caution as it can cause great confusion with end-users as there is no automatic on-screen indication that objects or expressions are in, or reference, alternate states. It is up to the QlikView developer to provide this information to the end-user using the **StateName()** function.

## Setting Alternate States

Do the following to enable and set alternate states:

1. Click **Alternate States...** in the **Document Properties: General** tab.  
The **Alternate States** dialog opens.

2. Click **Add...** to create a number of states, and name them. The names will be referred to as state identifiers.
3. Click **OK** to close the dialog.

Now the QlikView developer can create new Alternate States from within screen objects.

End-users who access QlikView documents from a QlikView Server can make use of Alternate States but cannot create Alternate States.

## Assigning States to Objects

Do the following to assign a state to an object:

1. Open the **Properties** dialog and select the **General** tab.
2. Set **Alternate State** to the state identifier you want to use for the object.

The object will now be independent from the rest of the document in terms of selection, unless other objects are set to the same state.

There are two states that are always available; **default state** and **inherited**. The **default state** is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**. Objects can inherit states from higher level objects, such as sheets and containers. This means that states are inherited as such: Document - Sheet - Sheet Objects. The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer.

## Comparing Alternate States

You can compare two states on the same object. For example, you can create a line chart to display both states of the data with the same axis by assigning a state in the expression using *Set Analysis (page 819)*.

Example:

If you have defined the states *Group1* and *Group2*, you can compare the average of field *Adj.Close* for the groups by showing these two expressions in a line chart:

```
Avg({ [Group1] } [Adj.Close])
```

```
Avg({ [Group2] } [Adj.Close])
```

## Logical Behavior when Using Alternate States

Pressing the **Clear** button affects all states.

In the Inter Explorer plugin, a menu item has been added to the **Clear** button drop-down menu that allows a user clear the selections of a specific state. This menu is not available in the Ajax client, but the same function can be achieved by creating an action button connected to the Clear action with a specific state.

Pressing the **Back** and **Forward** buttons affects all states. There is no mechanism to move back and forward through specific states.

Pressing the **Lock** and **Unlock** buttons affects all states. It is not possible to lock and unlock specific states with these buttons. It is possible to lock specific fields in states by using the right-click menu on list boxes in a specific state.

The menu items in the **Selections** menu apply to all states. There is no mechanism to affect specific states in this menu.

---

### Note!

**Alternate States** can be used with linked objects. The state applies to all instances of an object. Changing a state on a linked object will put the other linked objects into the same state.

---

### Note!

Variables belong to the default state. Changes in other states will not affect variable values.

---

### Note!

Triggers fire in all states.

---

**Note!**

Actions can be set to occur in specific states. One known exception is the **Run Macro** action. It can be set to run in a specific state; however macros fire in all states.

---

---

**Note!**

Objects that make use of a missing state (the state was removed by a developer) will revert to the default state though the **Alternate State** drop-down, that will display information such as:

AlternateStateName <unavailable>.

---

---

**Note!**

It is not possible to use chart field functions in conjunction with Alternate States as these functions do not have a `set_expression` parameter to set the state.

---



# 15 Bookmarks

## 15.1 Bookmark Types

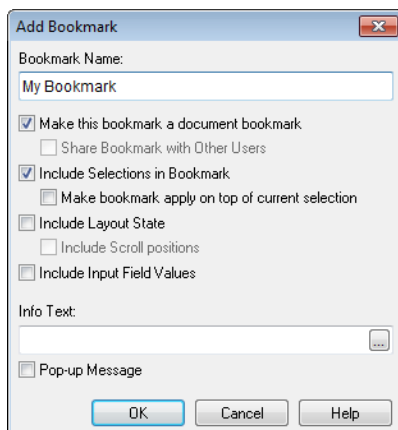
The current state of selections can be saved as bookmarks for later use. Bookmarks capture selections in all states defined in a QlikView document. When recalling a bookmark created in version 11 (or later) the selections in all states will be applied. There are different types of bookmarks:

- Document bookmarks** are stored inside the qvw document. They will always be available to whoever 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.

## 15.2 Add Bookmark Dialog

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 cause the **Add Bookmarks** dialog to appear.

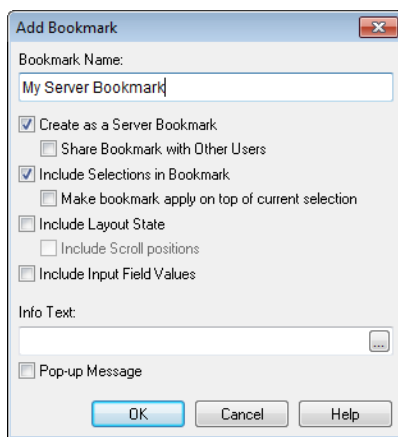
The dialog looks slightly different depending on whether it is a local document or a document on QlikView Server that the user is working with. We start by describing the dialog for local documents, and then describe the differences for server documents.



*The Add Bookmarks dialog for local documents*

<b>Bookmark Name</b>	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, it is possible to change the default name to a more explanatory text by entering an optional name.
<b>Make this bookmark a document bookmark</b>	By marking the 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 the computer.
<b>Include Selections in Bookmark</b>	By marking the check box the bookmark will include the selections made in the application.
<b>Make bookmark apply on top of current selection</b>	By marking the check box, the bookmark will be applied without first clearing existing selections in the document.
<b>Include Layout State</b>	<p>By marking the check box, the bookmark will store the layout state of all sheet objects on the active sheet at the time of the bookmark's creation. This includes, for example, 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.</p> <p>The layout state in bookmarks includes only show state (minimized/maximized) and expansion state, not the entire property of the object. Layout of all pivot tables on the active sheet will be preserved, unless they are located inside container objects.</p> <p>There is no reset option for pivot table layout. A workaround would be to create a document bookmark in the document clear state with the original layout of the pivot table.</p>
<b>Include Scroll Positions</b>	Marking this check box will include the current vertical scroll position in table objects in the bookmark.
<b>Include Input Field Values</b>	By marking this check box, the values in input fields will be stored in the bookmark.
<b>Info Text</b>	In the edit box it is possible to enter a text describing the bookmark or a message to be shown when recalling the bookmark.
<b>Popup Message</b>	If the 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:**



### Create as a server bookmark

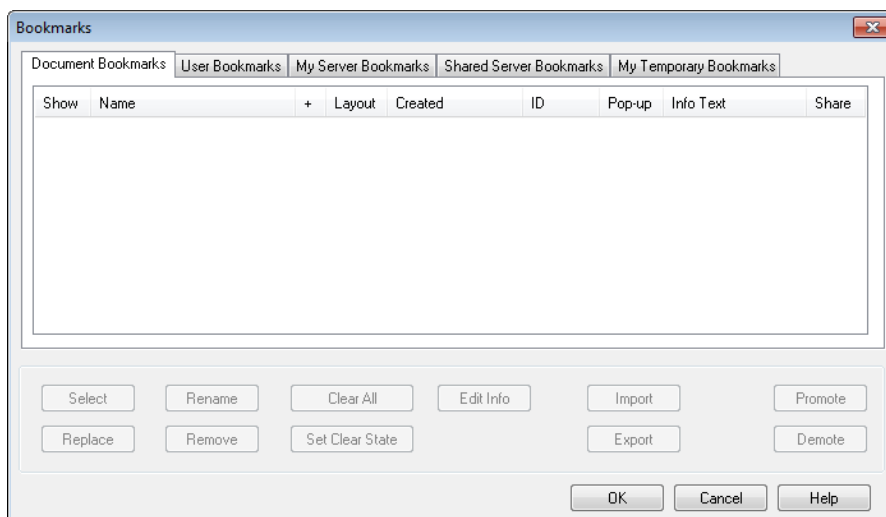
When this check box is marked, the bookmark will be created as a personal server bookmark, stored in a repository on the server. This option is available only if the user is an authenticated user and provided that the QlikView Server and the document owner allows the creation of server bookmarks.

If the check box is not marked, the bookmark will be created as a user bookmark, stored locally on the computer.

### Share Bookmark with Others

Provided that a personal server bookmark is created, marking this check box will immediately share the bookmark for use by other authenticated users of the same server document. It is possible at any time revoke the sharing by deselecting the **Share** check box in the **My Server Bookmarks** page of the **Bookmarks** dialog.

## 15.3 Bookmarks Dialog



*The Bookmarks dialog with five tabs, as it appears when working in a Server document*

The **Bookmarks** dialog can be opened by clicking **More...** in the **Bookmarks** menu. It is divided into two tabs (local documents) or five tabs (server documents), one for document bookmarks, one for user bookmarks, one for the personal server bookmarks (server documents only), one for other users' shared bookmarks (server documents only) and one for temporary bookmarks (server documents only). For more information about different types of bookmarks, see *Bookmark Types* (page 117).

At the top of the dialog a list of all bookmarks currently defined for the QlikView document are found. The bookmarks are described and may be sorted by several columns:

<b>Show</b>	Enable this option if the bookmark should appear in bookmark object listings and in the <b>Bookmarks</b> menu. If the check box is left unchecked, the bookmark will not appear in those places, but will remain accessible via this dialog.
<b>Name</b>	Name of the bookmarks. To sort the bookmarks in alphabetical order, click the header of the column. If clicked twice, the bookmarks are sorted in reverse alphabetical order.
<b>+</b>	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.
<b>Layout</b>	<p>If the bookmark was created complete with layout information, use the check-box to toggle the layout setting for the bookmark. When the layout setting is on, the program will attempt to recreate the layout that prevailed when the bookmark was created. This includes switching to the correct sheet and updating the layout of any sheet objects on that sheet.</p> <p>This setting cannot be used to add layout information to a bookmark that was originally created without it.</p>
<b>Created</b>	Timestamp information on when bookmarks were created.
<b>ID</b>	<p>The unique ID of the bookmark. Upon creation, every bookmark is assigned a unique ID for control via <i>Internal Macro Interpreter (page 893)</i>. The first bookmark of each type in a document will be assigned the ID BM01. This means it is possible to have a document bookmark and a server bookmark with the same ID.</p> <p>A bookmark ID may be changed via <i>Internal Macro Interpreter (page 893)</i> 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.).</p>
<b>Popup</b>	If info text has been added to the bookmark, checking this alternative shows it as a popup text.
<b>Info Text</b>	If info text has been added to the bookmark, it is displayed here.
<b>Share</b>	This check box is only available in the <b>My Server Bookmarks</b> page. By marking it for one of the personal server bookmarks, it makes that bookmark available to other authenticated users of the same server document in their <b>Shared Server Bookmarks</b> page. The bookmark will stay in the <b>My Server Bookmarks</b> page and will not appear in the <b>Shared Server Bookmarks</b> page. It is possible to revoke the sharing at any time by deselecting the check box.

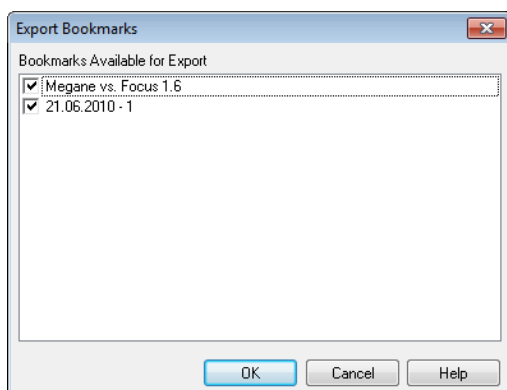
At the bottom of the dialog a number of buttons is found which can be used to perform actions with the bookmark currently selected in the list above.

<b>Select</b>	Displays the selected bookmark on the screen.
---------------	---



<b>Replace</b>	Replaces the selected bookmark with the current selections. Only bookmarks that you have created can be replaced.
<b>Rename</b>	Opens the <b>Rename Bookmark</b> dialog where it is possible to specify a new name for the bookmark. Only bookmarks that you have created can be renamed.
<b>Remove</b>	Removes the selected bookmark. Only bookmarks that you have created can be removed.
<b>Clear All</b>	Removes all bookmarks in the current pane of the dialog. Only bookmarks that you have created can be removed.
<b>Set Clear State</b>	Click this button if you want to use the selection of the highlighted bookmark as clear state.
<b>Edit Info</b>	Here it is possible to edit the info text.
<b>Import</b>	After browsing for and selecting a previously saved bookmark (.qbm) file, the <i>Import Bookmarks</i> (page 122) dialog will open to import bookmarks.
<b>Export</b>	Opens the <i>Export Bookmark(s)</i> (page 121) dialog where it is possible to export selected bookmarks to a QlikView bookmark (.qbm) file.
<b>Promote</b>	Moves the selected bookmark one step up in the list. It is also possible to do this by clicking and dragging it to any position in the list.
<b>Demote</b>	Moves the selected bookmark one step down in the list.
<b>Move Local User Bookmarks to Server</b>	This text only appears when working with a server document and in the <b>User Bookmarks</b> tab. By clicking on the text it is possible to convert all the local user bookmarks to server bookmarks, stored in the server repository. This is a recommended practice as the personal server bookmarks will stay accessible via the server even if changing computer or if the server document is renamed. Before the conversion takes place, the system asks for confirmation. The conversion is all-or-nothing (within the active document) and one-way only.

## 15.4 Export Bookmark(s)



*The Export Bookmarks dialog*

In the **Export Bookmarks** dialog a list of all bookmarks that have been defined in the current document, is found. Only those bookmarks that are ticked will be included in the exported bookmark (.qbm) file.

On clicking **OK** the specification of a file name and location for the bookmark file will be requested.. Once saved, the bookmark file can be kept for later use or distributed to other users of the same QlikView document.

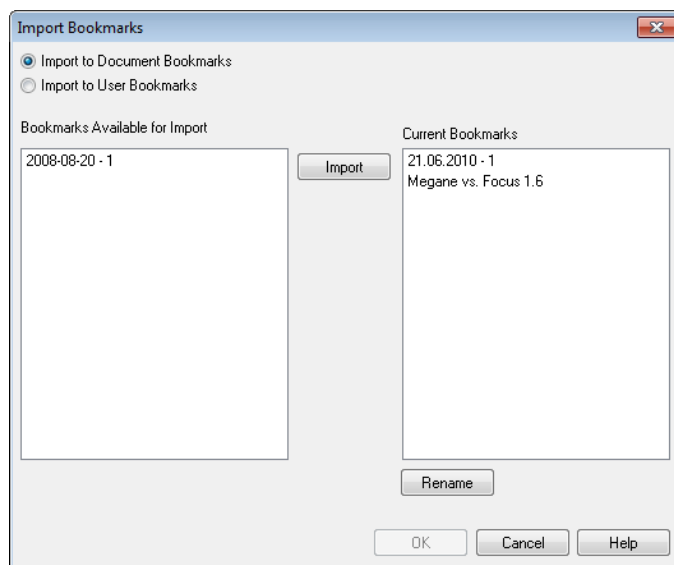
## 15.5 Import Bookmarks

Bookmarks are imported to a bookmark file by choosing **Import...** from the **Bookmarks** menu. It opens a browse dialog where a QlikView bookmark file (.qbm extension) can be selected. The **Import Bookmark** dialog will then open.

In the **Import Bookmarks** dialog, a list of all bookmarks that are included in the selected bookmark file, are found.

### Note!

Bookmarks should only be imported to a QlikView document containing fields and field values to which the bookmarks refer.



*The Import Bookmarks dialog*

### **Bookmarks Available for Import**

To the left all the bookmarks which were defined in the bookmark file are listed.

### **Current Bookmarks**

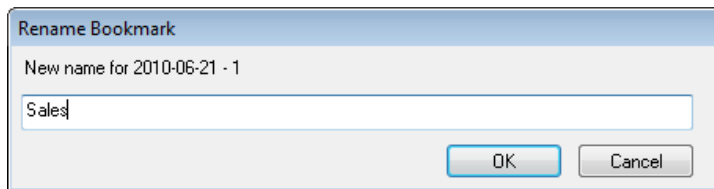
To the right a list of existing user bookmarks or document bookmarks in the document is shown. Bookmarks must be imported either as a user bookmark or as a document bookmark. The radio buttons at the top of the dialog governs how the bookmark is imported.

### **Import**

Highlight one or more bookmarks in the **Bookmarks Available for Import** list and press this button to import them.

### **Rename**

Opens the **Rename Bookmark** dialog where a new name for the highlighted bookmark can be specified before import.



## 15.6 Alternate States and Bookmarks

Bookmarks capture selections in all states defined in the QlikView document. When recalling a bookmark created in version 11 (or later) the selections in all states will be applied.

It is possible to make use of specific states contained in a bookmark within an expression. For example, this expression calculates Sales over the set generated by the selections defined in bookmark BM01 for the state 'Group 1'.

### Example:

```
sum({[Group 1]::BM01} Sales)
```

---

#### Note!

Existing bookmarks in QlikView documents created prior to version 11 will contain the **default state** only when used in QlikView 11. Bookmarks that are created or modified in version 11 will contain all states. When these bookmarks are used in a version prior to 11, the additional states are ignored.

---

#### Note!

Bookmarks that reference a state that no longer exists (the state was removed by a developer) will ignore the missing state(s).

---



---

# 16 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.

- |                              |  |
|------------------------------|--|
| <b>Document Reports</b>      | 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.  |
| <b>User Reports</b>          | 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 (that is, not from AJAX clients). |
| <b>My Server Reports</b>     | 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.  |
| <b>Shared Server Reports</b> | 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.

---

## 16.1 Reports Menu

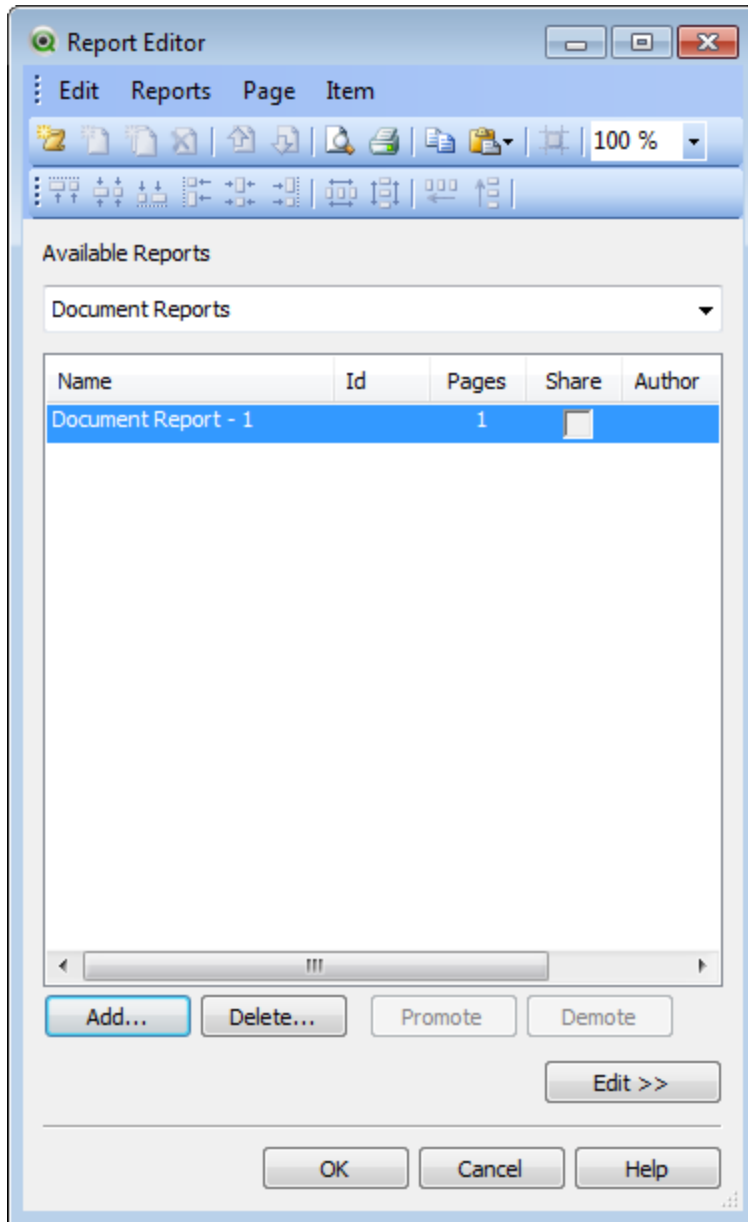
The **Reports** menu at the top of the QlikView screen contains the command **Edit Reports...** that opens the *Report Editor* (page 126) dialog where new reports can be created or existing reports edited. From this dialog it is also possible to delete and select reports, design their layout, add pages and images etcetera. The report editor lists all available reports for immediate selection.

Reports may be **Document Reports**, stored with the document, or **User Reports**, stored separately on the user’s computer. See *Report Editor* (page 126).

## 16.2 Report Editor

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



*The Report List page of the Report Editor dialog*

The **Report List** page is used to manage reports. It is possible to create new reports and remove existing reports. This is also where a report is selected for page editing in the **Page Editor**. At the top of the page a drop-down and a list of reports are found. The drop-down is used for:

**Available Reports****Document Reports**

Opens 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.

**User Reports**

Opens a list of all user reports, stored separately on the user's computer. Clicking one of the report names opens the **Print** dialog for printing of the report.

**My Server Reports**

Personal 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 personal server reports shared server reports are stored in a repository on the server.

Reports may be **document reports**, stored with the document, or **user reports**, stored separately on the user's computer. In the drop-down list **Available Reports** at the top of the Report Editor screen you select whether **Document Reports** or **User Reports** should be shown in the list. For server documents you may choose between **My Server Reports**, **Shared Reports** or **User Reports**. The list itself includes several columns:

<b>Name</b>	Name of the report.
<b>ID</b>	The unique report ID (see below).
<b>Pages</b>	Number of pages currently defined in the report.
<b>Share</b>	This check box is only available in the My Server Reports list. By marking it for one of the My Server Reports, the report will be available to other authenticated users of the same server document in their Shared Server Reports list. The report will stay in the My Server Reports list and will not appear in the Shared Server Reports list. Deselect the check box at any time to revoke the sharing.
<b>Author</b>	The authenticated name of the creator of the report.
<b>Add...</b>	Push this button to create a new report. If <b>Document Reports</b> is selected above the <b>List of Reports</b> , the new report will be a document report. If User Reports is selected, the new report will be a user report. The dialog <b>New Report</b> is opened so that you can name your report. Check <b>Copy Pages from Other Report</b> and pick an existing report in the drop-down menu to copy pages from that report.
<b>Delete</b>	Push this button to delete the report currently selected in the <b>List of Reports</b> .
<b>Promote</b>	Push this button to promote the report currently selected in the <b>List of Reports</b> one step up in the list.
<b>Demote</b>	Push this button to demote the report currently selected in the <b>List of Reports</b> one step down in the list.

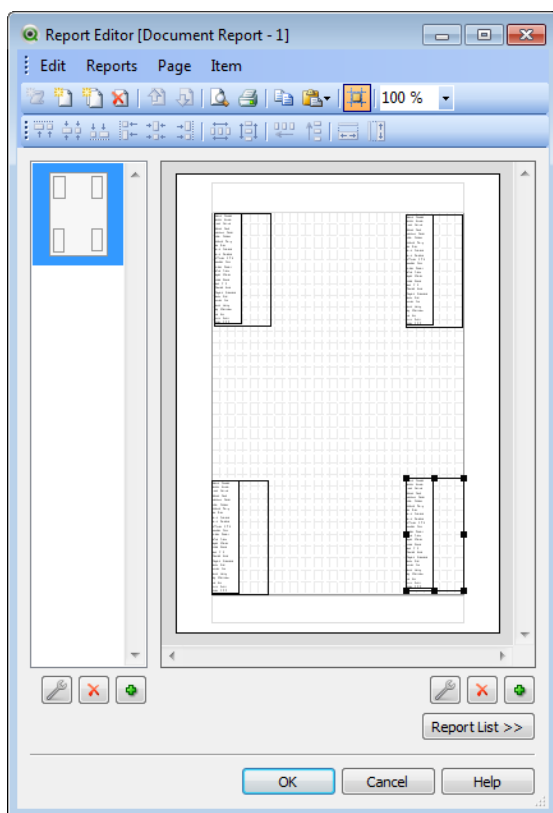
**Move Local User Reports to Server**

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 the local user reports to server reports, stored in the server repository. This is a recommended practice as the personal server reports will stay accessible via the server even if the computer is changed or if the server document is renamed. Before the conversion takes place, the system asks for confirmation. 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 on a report in the **List of Reports**.

## Page Editor



*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.



## Edit Menu

<b>Copy</b>	Copies the currently selected item and the current page in the report. Also available as right-click command in the page preview pane.
<b>Paste Report</b>	Pastes the report from the Clipboard.
<b>Paste Page</b>	Pastes the page from the Clipboard.
<b>Paste Item</b>	Pastes the item from the Clipboard. Also available as right-click command in the page preview pane.
<b>Snap to Grid</b>	Displays a grid in the report to help align the report items.

## Reports Menu

<b>Add</b>	Adds a report. Only available when viewing the <b>Report List</b> .
<b>Delete</b>	Deletes a report. Only available when viewing the <b>Report List</b> .
<b>Promote</b>	Moves the selected report up one step in the list of reports. Only available when viewing the <b>Report List</b> .
<b>Demote</b>	Moves the selected report down one step in the list of reports. Only available when viewing the <b>Report List</b> .
<b>Export</b>	Saves the report file as an XML document. Only available when viewing the <b>Report List</b> .
<b>Import</b>	Imports a report that has been saved as XML. Only available when viewing the <b>Report List</b> .
<b>Print Preview...</b>	Opens the <b>Print Preview</b> dialog and where you can see how the current report will look when printed with the current selections in the QlikView document.
<b>Print...</b>	Opens the <b>Print</b> dialog and prints the current report.
<b>Report Settings</b>	Opens the <i>Report Editor: Report Settings Dialog (page 133)</i> dialog where you can set various properties for the report. This includes margins, header/footer setting etc.

## Page Menu

<b>Promote</b>	Promotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.
<b>Demote</b>	Demotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.
<b>Add Multi Page</b>	Adds a multi-paper page after the currently selected page.
<b>Add Single Page</b>	Adds a single-paper page after the currently selected page.
<b>Delete</b>	Deletes the currently selected page.
<b>Page Settings</b>	Opens the <i>Report Editor: Page Settings Dialog (page 136)</i> dialog where you can set various properties for the selected page.

## Item Menu

<b>Left Align</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the left. Also available as right-click command in the page preview pane.
<b>Center Horizontally</b>	Only available when two or more print objects are selected. Centers the selected objects horizontally. Also available as right-click command in the page preview pane.
<b>Right Align</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the right. Also available as right-click command in the page preview pane.
<b>Bottom Align</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the bottom. Also available as right-click command in the page preview pane.
<b>Center Vertically</b>	Only available when two or more print objects are selected. Aligns the selected objects along their center on the vertical axis. Also available as right-click command in the page preview pane.
<b>Top Align</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the top. Also available as right-click command in the page preview pane.
<b>Space Horizontally</b>	Only available when two or more print objects are selected. Distributes the selected objects on the horizontal axis with equal spaces between them. Also available as right-click command in the page preview pane.
<b>Space Vertically</b>	Only available when two or more print objects are selected. Distributes the selected objects on the vertical axis with equal spaces between them. Also available as right-click command in the page preview pane.
<b>Adjust Left</b>	Only available when two or more print objects are selected. Arranges the active sheet objects from the vertical edge of the left-most object and to the right with minimal spaces between them. Also available as right-click command in the page preview pane.
<b>Adjust Top</b>	Only available when two or more print objects are selected. Arranges the active sheet objects from the horizontal top edge of the topmost object and downwards with minimal spaces between them. Also available as right-click command in the page preview pane.
<b>Same Width</b>	Only available when two or more print objects are selected (with the help of Shift-click). Gives the currently selected items the same width. Also available as right-click command in the page preview pane.
<b>Same Height</b>	Only available when two or more print objects are selected (with the help of Shift-click). Gives the currently selected items the same height. Also available as right-click command in the page preview pane.
<b>New Image...</b>	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.

<b>New Text...</b>	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.
<b>New Current Selections...</b>	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.
<b>Delete</b>	Deletes the currently selected item(s). Also available as right-click command in the page preview pane.
<b>Item Settings...</b>	Opens the <i>Report Editor: Item Settings Dialog (page 139)</i> dialog where you can set various properties for the selected print item. Also available as right-click command in the page preview pane.

At the bottom of the two panes there are icons for some of the functions in the different menus:

<b>Page Settings</b>	Opens the <i>Report Editor: Page Settings Dialog (page 136)</i> dialog where you can set various properties for the selected print item. Also available as right-click command in the page preview pane.
<b>Delete</b>	Deletes the highlighted page.
<b>Add</b>	Adds a single-paper page or a multi-paper page after the currently selected page.
<b>Item Settings</b>	Opens the <i>Report Editor: Item Settings Dialog (page 139)</i> dialog where you can set various properties for the selected print item. Also available as right-click command in the page preview pane.
<b>Delete</b>	Deletes the highlighted item.
<b>Add</b>	Adds a new image, new text or a new current selections stamp.

## Page List Pane

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 page 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 a 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 it is possible to 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

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:

<b>Add Single Paper Page</b>	Adds a single paper page after the currently selected page.	
<b>Add Multi Paper Page</b>	Adds a multi paper page after the currently selected page.	
<b>Delete Page</b>	Deletes the currently selected page.	
<b>Promote Page</b>	Promotes the selected page one step up the list. It is also possible to drag and drop the page in the page list pane.	
<b>Demote Page</b>	Demotes the selected page one step down the list. It is also possible to drag and drop the page in the page list pane.	

## Page Preview Pane

To the right of the page list pane a preview of the page currently selected is found in the page list pane.

### Adding sheet objects to the report

Simply drag or double-click on any sheet object in the layout to put it in 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!






### Sizing and positioning of 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.

### Navigating and selecting print objects on page

Select a print object by clicking on it. Once one print object has been selected, it is possible to move the selection to the next object using the Tab key (Shift+Tab to move backwards). Use Shift-click to select multiple objects. In multi paper pages, drag print objects between the different areas.


When one or two items are highlighted in the preview pane, the commands in the align toolbar become available.

<b>Align Top</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the top.	
<b>Center Vertically</b>	Only available when two or more print objects are selected. Aligns the selected objects along their center on the vertical axis. Also available as right-click command in the page preview pane.	
<b>Align Bottom</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the bottom.	
<b>Align Left</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the left.	
<b>Center Horizontally</b>	Only available when two or more print objects are selected. Centers the selected objects horizontally. Also available as right-click command in the page preview pane.	





<b>Align Right</b>	Only available when two or more print objects are selected (with the help of Shift-click). Aligns the selected objects to the right.	
<b>Space Horizontally</b>	Only available when two or more print objects are selected. Distributes the selected objects on the horizontal axis with equal spaces between them. Also available as right-click command in the page preview pane.	
<b>Space Vertically</b>	Only available when two or more print objects are selected. Distributes the selected objects on the vertical axis with equal spaces between them. Also available as right-click command in the page preview pane.	
<b>Adjust Left</b>	Only available when two or more print objects are selected. Arranges the active sheet objects from the vertical edge of the left-most object and to the right with minimal spaces between them. Also available as right-click command in the page preview pane.	
<b>Adjust Top</b>	Only available when two or more print objects are selected. Arranges the active sheet objects from the horizontal top edge of the topmost object and downwards with minimal spaces between them. Also available as right-click command in the page preview pane.	

## **Toolbar commands**

There are two toolbar commands directly associated with the page preview pane:

<b>Design Grid</b>	Toggles the design grid on and off. This also activates the snap to grid functionality. With design grid and snap-to-grid activated it is easier to align the sheet objects on the page, but you have less exact control.	
<b>Zoom</b>	In this drop-down, specify a zoom factor for the entire preview pane. With a larger zoom factor it becomes much easier to do exact placement of objects.	

**Then, there are some toolbar buttons for printing the report from inside the Report Editor dialog.**

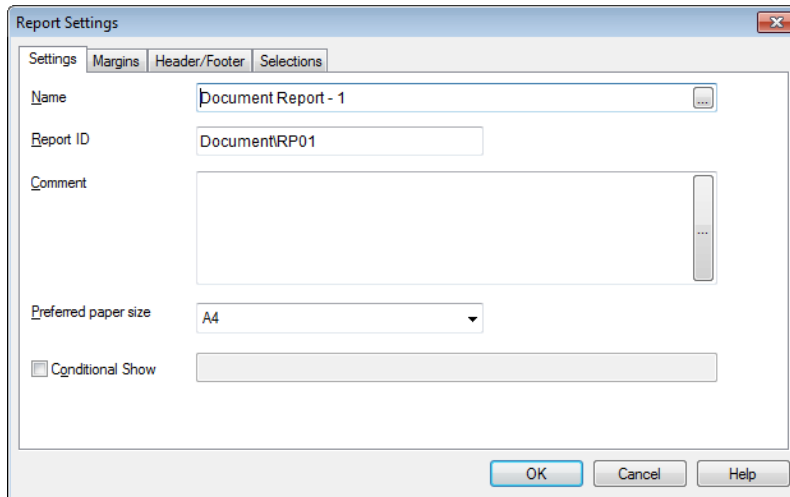
<b>Print Preview</b>	Push this button to open the <b>Print Preview</b> dialog and see how the current report will look when printed with current selections in the Qlik-View document. This button is also available from the <b>Page Editor</b> page of the <b>Report Editor</b> dialog.	
<b>Print</b>	Push this button to open the Print dialog and print the current report. This button is also available from the <b>Page Editor</b> page of the <b>Report Editor</b> dialog.	
<b>Copy</b>	Copies the highlighted report, page or item.	
<b>Paste</b>	Pastes the copied report, page or item.	

Finally, there is a toolbar button for pasting copied reports, pages and items.

## **Report Editor: Report Settings Dialog**

In this dialog you will find global (non page related) settings for the report. The dialog has four pages.

## The Settings Tab

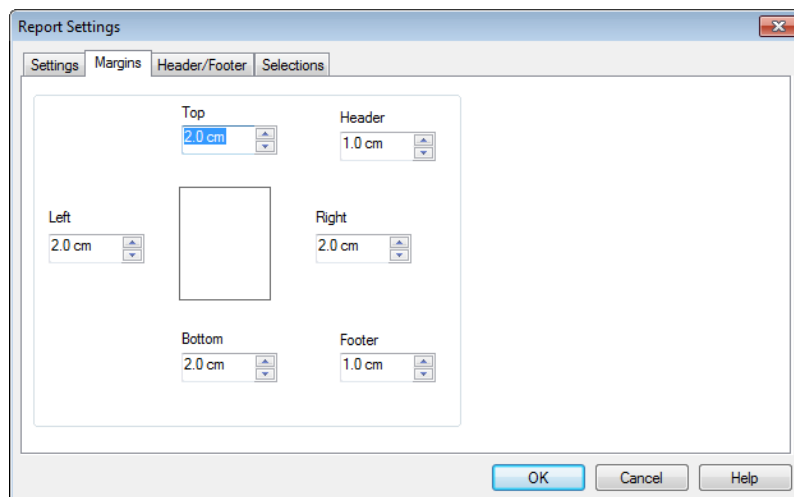


*Report Settings, Settings tab*

This tab contains basic properties of the report:

- |                             |  |
|-----------------------------|--|
| <b>Name</b>                 | Name of the report. This may be any text string. The name may be given as a calculated formula for dynamic update.   |
| <b>Report Id</b>            | 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..   |
| <b>Comment</b>              | 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  |
| <b>Preferred Paper Size</b> | In this drop-down you can specify the size of paper intended for the report. The proportions of the 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.   |
| <b>Conditional Show</b>     | 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 <b>Show All Sheets and Sheet Objects</b> in the dialog <i>Document Properties: Security (page 438)</i> . This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+S. |

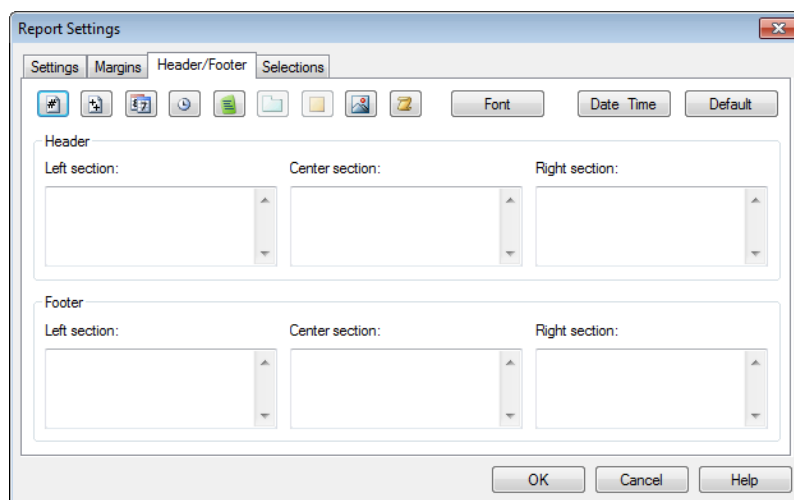
## The Margins Tab



*Report Settings, Margins tab*

This tab contains margin settings for the report. The controls are the same as for the *Print: Layout (page 92)* page of the **Print** dialog.

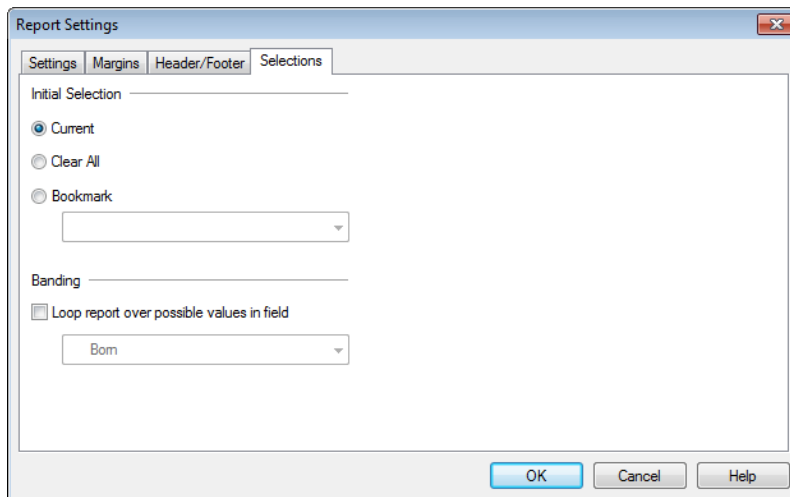
## The Header/Footer Tab



*Report Settings, Header/Footer tab*

This tab contains header/footer settings for the report. The controls are the same as for the *Print: Header/Footer (page 93)* page of the **Print** dialog.

## The Selections Tab



### *Report Settings, Selections tab*

This tab contains settings for the selections to be used when printing the report:

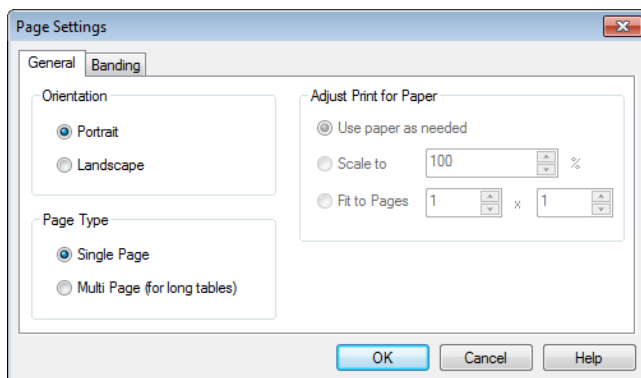
- |  |  |
|--|--|
| <b>Initial Selections</b>                        | These radio buttons allow you to state the initial selections for the report print. Regardless of which setting you use, the selections prevailing before the report print will be re-established after the print.   |
| <b>Current Selections</b>                        | Use current selections as the basis of the report print (default).   |
| <b>Clear All</b>                                 | All current selections in the document will be cleared before printing the report. After completed printing, the original selections will be re-applied.   |
| <b>Current Selections</b>                        | 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 re-applied.   |
| <b>Loop Report Over Possible Values in Field</b> | 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 re-applied. If page numbers are used, these will be continuous over all pages printed. |

## Report Editor: Page Settings Dialog

In this dialog the settings for the currently selected page are found. The dialog has three pages.



## General

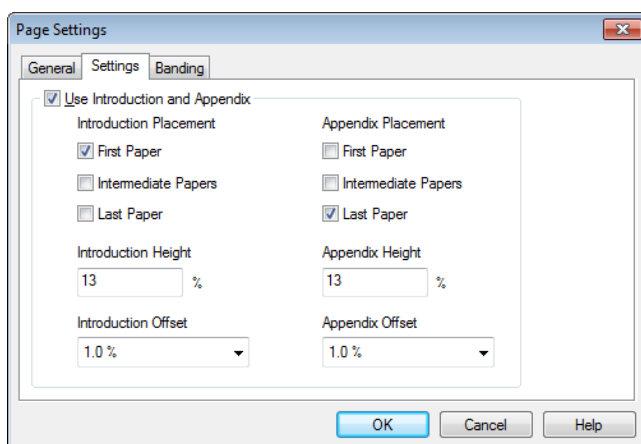


### *Page Settings, General tab*

This tab contains basic properties of the page:

- |                               |   |
|-------------------------------|---|
| <b>Orientation</b>            | Each page can have its separate setting for orientation. Choose between <b>Portrait</b> and <b>Landscape</b> . The choice will be reflected in the page list pane above.  |
| <b>Page Type</b>              | Set if the report should be printed as a <b>Single Page</b> or as a <b>Multi Page</b> (for long tables).  |
| <b>Adjust Print for Paper</b> | This setting is only available for multi paper pages. <ul style="list-style-type: none"> <li><b>Use Paper as Needed</b><br/>No scaling of the sheet object printout. As many pages as needed in each direction will be used.</li> <li><b>Scale to x %</b><br/>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.</li> <li><b>Fit to x by y</b><br/>The sheet object printout will be scaled to fit on a fixed number of pages.</li> </ul> |

## Settings

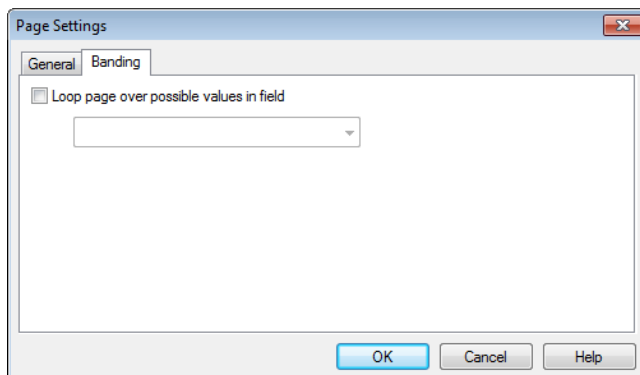


### *Page Settings, General tab*

This tab contains settings for introduction and appendix areas in multi-paper pages. It is not available for single-paper pages.

<b>Use Introduction and Appendix</b>	Enable this option to enable the use of fixed size print items in addition to the dynamic size content on a multi-paper page. When doing this you will unfortunately lose backward compatibility with QlikView versions prior to 7.5.
<b>Introduction Placement</b>	By marking one, two or three of the check boxes you can choose to print the introduction area on the <b>First Page</b> , on all <b>Intermediate Pages</b> and/or on the <b>Last Page</b> .
<b>Introduction Height</b>	Defines the height assigned on each page for the introduction area. The number is given in % of available print area (paper height minus header/footer areas). This setting may also be changed by dragging the border directly in the page preview pane.
<b>Introduction Offset</b>	Defines the spacing between the introduction area and the main area. the value is given in % of available print area (paper height minus header/footer areas).
<b>Appendix Placement</b>	By marking one, two or three of the check boxes you can choose to print the appendix area on the <b>First Page</b> , on all <b>Intermediate Pages</b> and/or on the <b>Last Page</b> .
<b>Appendix Height</b>	Defines the height assigned on each page for the appendix area. The number is given in % of available print area (paper height minus header/footer areas). This setting may also be changed by dragging the border directly in the page preview pane.
<b>Appendix Offset</b>	Defines the spacing between the introduction area and the main area. the value is given in % of available print area (paper height minus header/footer areas).

## Banding



*Page Settings, Banding tab*

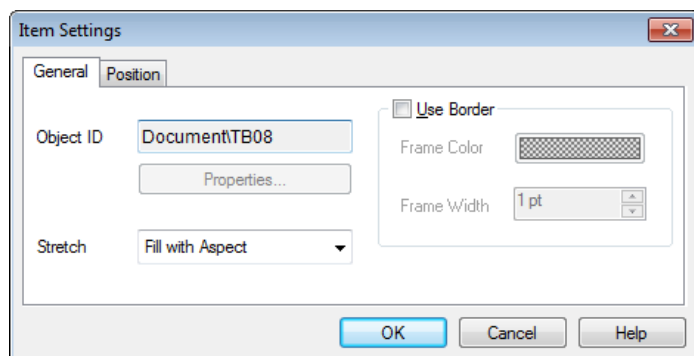
This tab contains properties for banded output of the page. This means that the page is printed repeatedly for each possible value in a specified field.

When the check box **Loop Page Over Possible Values in Field** 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 *The Selections Tab (page 136)*) 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 re-applied. If page numbers are used, these will be continuous over all pages printed.

## Report Editor: Item Settings Dialog

In this dialog you will find settings for the print item (object) currently selected in the page preview pane. The dialog has two pages.

### General

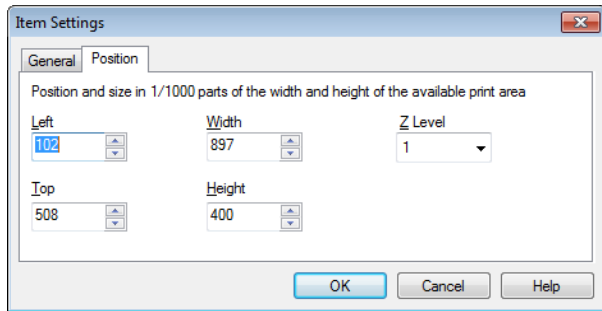


*Item Settings, General tab*

The **General** tab contains settings for the print item currently selected in the page preview pane.

<b>Object ID</b>	The sheet object ID of the sheet object to be printed.
<b>Properties...</b>	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 reflected in the report print result.
<b>Stretch</b>	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.
<b>Clip</b>	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.
<b>Fill</b>	Size the print object so that it fills the placeholder rectangle.
<b>Fill with Aspect</b>	Size the print object so that it fits in the placeholder rectangle while keeping the original sheet object aspect ratio.
<b>Use Border</b>	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.
<b>Frame Color</b>	Press this button to select the border color.
<b>Frame Width</b>	Here you can specify the thickness of the border.

## Position



*Item Settings, Position tab*

The **Position** tab contains settings for positioning and sizing the objects in the available print area.

- |                |   |
|----------------|---|
| <b>Left</b>    | Sets the position of the currently selected item(s) relative to the left margin. 0 is to the far left of the available print area.  |
| <b>Top</b>     | Sets the position of the currently selected item(s) relative to the top. 0 is at the top of the available print area.   |
| <b>Width</b>   | Sets the size of the currently selected item(s) in 1/1000 parts of the width of the available print area.   |
| <b>Height</b>  | Sets the size of the currently selected item(s) in 1/1000 parts of the height of the available print area.  |
| <b>Z Level</b> | 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. |

# 17 Alerts

## 17.1 Creating Alerts

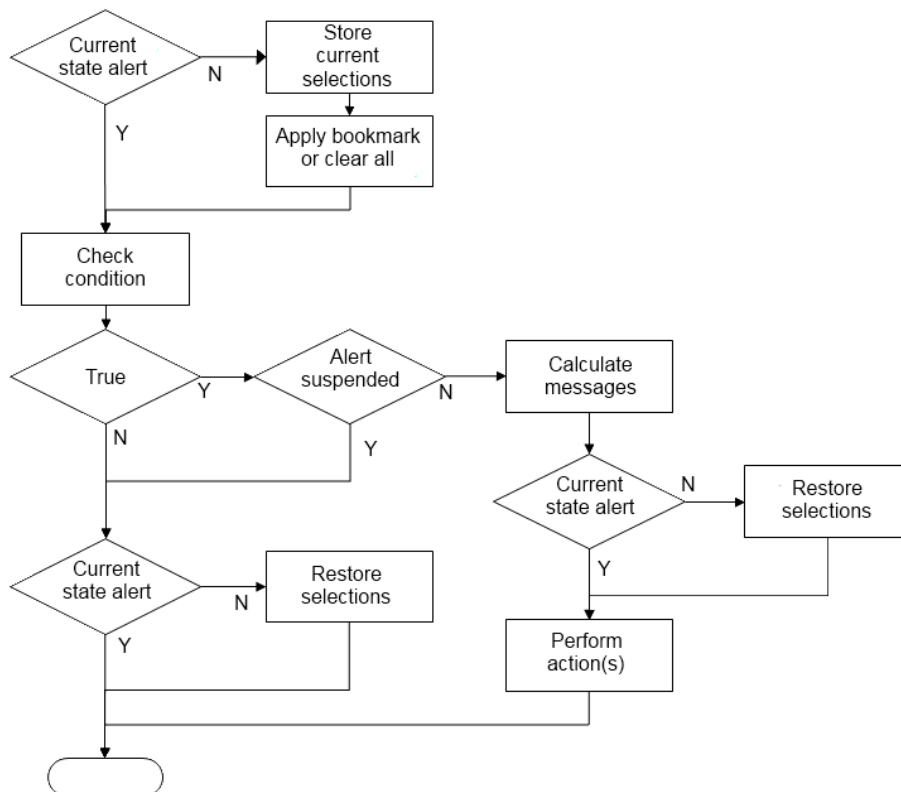
The easiest way to define an alert is by using the **Alert Wizard**, which can be invoked from the **Tools** menu. Alerts may also be created and maintained in the **Alerts** dialog, which can be opened from the **Tools** menu. Alerts are stored as part of the QlikView document. Alerts can only be created and triggered from Windows versions of QlikView (that is, not from AJAX clients).

## 17.2 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.

Opens the *Alerts (page 142)* dialog where all QlikView alerts are created and maintained.

## 17.3 Alerts

The easiest way to define an alert is by using the *Alert Wizard (page 146)* that can be invoked from the **Tools** menu.

The **Alerts** dialog is opened from the **Tools** menu. This dialog is used for managing alerts, which are stored as part of the QlikView document. Alerts can be triggered from any Windows version of QlikView (that is, not from AJAX clients).

An alert is a composite entity typically consisting of three basic parts:

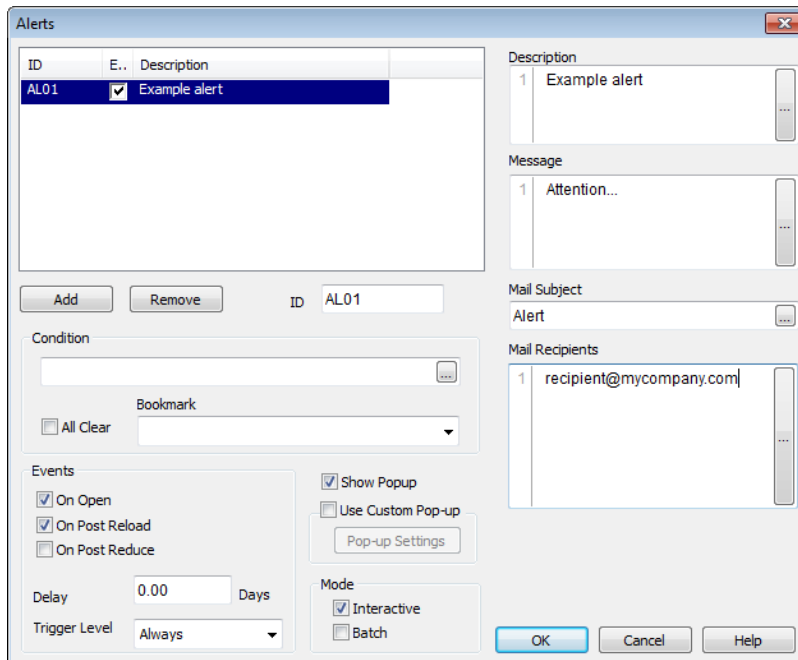
1. A condition, i.e. a QlikView expression forming a logical condition, that can be either true or false.
2. A logical state (bookmark, clear all or current selection state) that should be applied before checking the state of the condition expression.
3. One or more actions to be performed when the condition 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 and the condition is met and the action(s) performed, it can be said that to have *fired*. QlikView alert checks can be triggered in three different ways:

- a. 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.
- b. Manually from a *Internal Macro Interpreter (page 893)* via special Automation APIs. Refer to the QlikView file *APIguide.qvw* (a very useful QlikView file describing the use of the macro functionality in QlikView, which is normally installed with the program) for details.
- c. Externally from programs running QlikView in batch mode which have a special Automation API to retrieve a list of fired alerts from a given context.

Use caution when creating alerts, large amounts of macro-triggered alerts can make the document sluggish!

## 17.4 The Alerts Dialog



### *The Alerts dialog*

The top left part of the dialog contains a list of alerts where all alerts currently defined in the document are displayed. The list includes three columns: **ID**, **Enable** and **Description**.

**ID** The unique ID of the alert is displayed in the **ID** edit box as well as in the list of alerts. Upon creation, every QlikView layout entity, including alerts, is assigned a unique ID for control via *Internal Macro Interpreter* (page 893). The first alert of a document will be assigned the ID AL01. It is possible to edit this ID number later on in the **ID** edit box.

**Enable** Select or deselect the check box in order to enable/disable the alert.

**Description** Description of an alert as defined under **Description** (see below).

**Add** The **Add** button adds a new default alert to the list for further editing.

**Remove** Selecting an alert and clicking the **Remove** button deletes it from the list.

**Condition**

The **Condition** group is used for defining the conditional expression. (**Condition**)

This is a QlikView expression that should evaluate to true (non zero) when the alert is to fire.

**All Clear**

If this option is checked the all clear state will be used when evaluating the alert condition.

**Bookmark**

If **All Clear** is left unchecked, 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 (i.e. the logical state of the document) will be used when the alert condition is tested.



**Events**

The **Events** group is used for defining automatic event checks.

**On Open**

The alert is automatically checked when the document is opened.

**On Post Reload**

The alert is automatically checked after a script execution (**Reload**) of the document. In addition, the alert is also checked upon opening of the document, provided that the last reload time stamp stored in the document is later than the time stamp noted when the alert was last checked.

**On Post Reduce**

The alert is automatically checked after a data reduction (**Reduce Data** command) of the document. In addition, the alert is also checked upon opening of the document, provided that the last reduction time stamp stored in the document is later than the time stamp noted when the alert was last checked.

**Delay**

The re-firing of automatically checked alerts may be suspended for a given number of **Days** after an alert has been fired. Decimals may be used for specifying fractions of a day. The value 0 of course indicates that no delay 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 the drop-down list you can choose between three levels of reoccurrence:

**Always**

means that the alert is fired every time the trigger event(s) occur and the alert condition is fulfilled.

**Message Changes**

means that 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** line (see below) will be regarded as change of message.)

**State Changes**

means that the firing of the alert is suspended until the alert state changes, i.e. the alert condition was unfulfilled during at least one check and then becomes fulfilled again during a later check. This is the strongest type of alert suspension.

**Show Popup**

Mark the check box **Show Pop-up** to show the alert message as a pop-up balloon when the alert is fired.

**Use Custom Popup**

The **Use Custom Pop-up** group lets you define a custom pop-up format for the selected alert. By checking this alternative, you can replace the default **Alert Pop-up Settings** as defined in *Document Properties: General (page 424)*. The **Pop-up Settings** button opens the *Pop-Up Window Settings (page 427)* dialog.

<b>Mode</b>	In the <b>Mode</b> group you can check the relevant options in order to define an alert as an <b>Interactive</b> (automatic triggers in layout) and/or as an alert relevant for external programs running QlikView in <b>Batch</b> mode (command line execution), e.g. QlikView Publisher, via the special Automation API for Batch triggers. If none of the check boxes in this group are marked, the alert can still be manually checked via macros.
<b>Description</b>	A commentary field where the creator of an alert can describe the purpose of an alert. It is only used in the list of alerts in this dialog.
<b>Message</b>	The <b>Message</b> edit box is where you type the message to be displayed with the alert. For pop-up alerts the text will appear in the pop-up, for e-mail alerts this is the body text of the e-mail. The message text may be defined as a <i>Calculated Formula (page 871)</i> for dynamic update. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas.
<b>Mail Subject</b>	In the <b>Mail Subject</b> edit box a text to be used for the subject line in e-mail alert messages can be entered. The text may be defined as a calculated label expression for dynamic update. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas. If left empty, the subject line will display the text 'QlikView Alert'.
<b>Mail Recipients</b>	The <b>Mail Recipients</b> pane is a semi-colon separated list of mail addresses. Each addressee will receive an e-mail message whenever the alert fires. The list may be defined as a calculated string expression for dynamic update.

## 17.5 Alert Wizard

The alert wizard helps with the task of defining an alert.

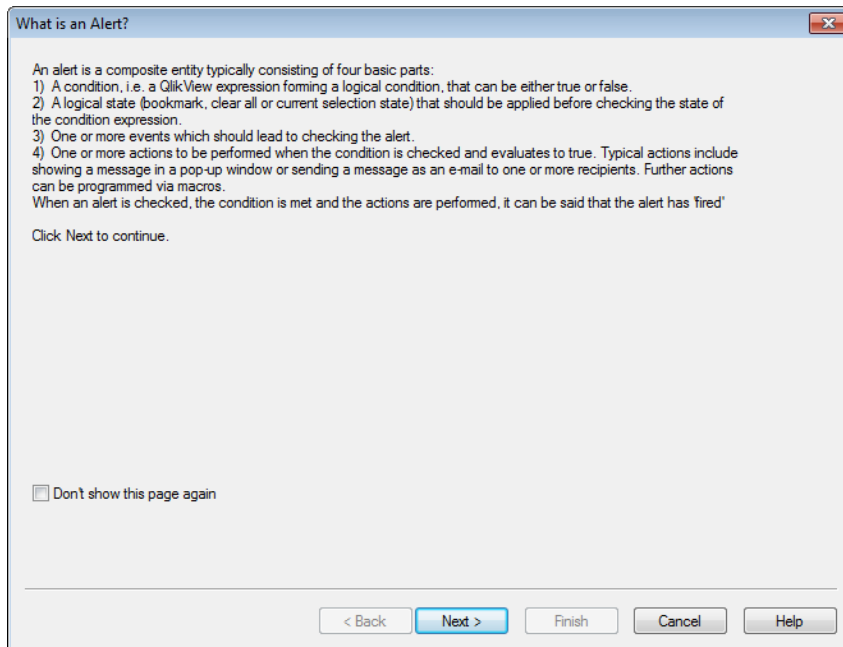
The process has the following basic steps:

<b>Step 1 - Name/Describe the Alert</b>	<i>page 148</i>
<b>Step 2 - Define the Alert Condition</b>	<i>page 148</i>
<b>Step 3 - Define when to Test the Alert Condition</b>	<i>page 149</i>
<b>Step 4 - Define Test Delay</b>	<i>page 150</i>
<b>Step 5 - Define the Action(s) to Take when the Alert Fires</b>	<i>page 151</i>

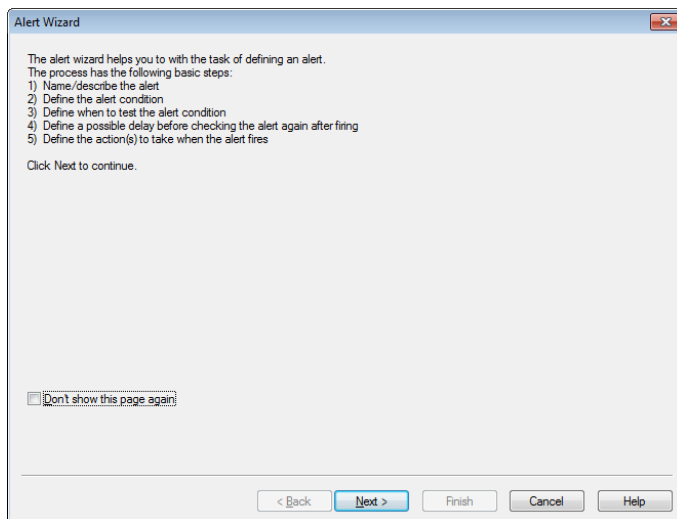
Depending on selections, some of the above steps may be skipped.

### Starting the alert wizard

Start the alert wizard by choosing **Alert Wizard** from the **Tools** menu.



*The first start page of the Alert Wizard*



*The second 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.

## Step 1 - Name/Describe the Alert

*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

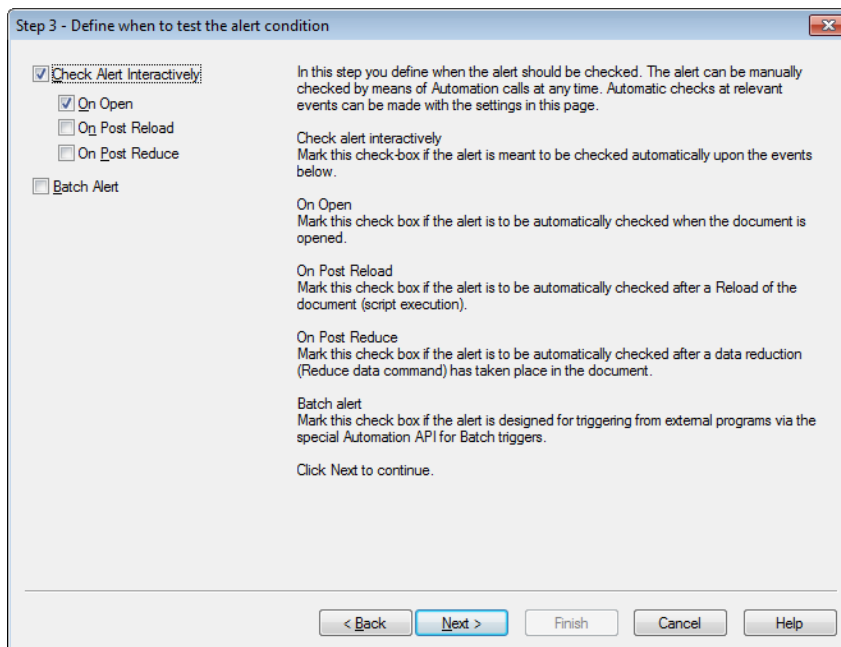
*The second page of the Alert wizard*

In this step you define the alert condition and the selection state that should be used when checking the alert condition.

<b>Alert condition</b>	Enter a QlikView expression that should evaluate to true (non zero) when the alert is to fire.
<b>Selection state when testing alert</b>	<p>Here you decide which selection state should be used when testing the result of the alert condition. There are three choices:</p> <p><b>Current Selection</b> The alert condition will be tested using whatever selections are made at the time of checking the alert.</p> <p><b>All Clear</b> All selections will be temporarily cleared before testing the alert condition.</p> <p><b>Bookmark</b> A specified bookmark will temporarily be applied before testing the alert condition.</p>

Click **Next** to continue.

### Step 3 - Define when to Test the Alert Condition



*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 inter-actively**

Enable this option if the alert is meant to be checked automatically upon the events below.

**On Open**

Enable this option if the alert is to be automatically checked when the document is opened.

**On Post Reload**

Enable this option if the alert is to be automatically checked after a Reload of the document (script execution).

**On Post Reduce**

Enable this option if the alert is to be automatically checked after a data reduction (Reduce data command) has taken place in the document.

**Batch alert**

Enable this option 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

Trigger Level Always

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 then becomes fulfilled again during a later check. This is the strongest type of alert suspension.

Click Next to continue.

< Back   Next >   Finish   Cancel   Help

*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 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**  
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**  
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.

**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.

**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

*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

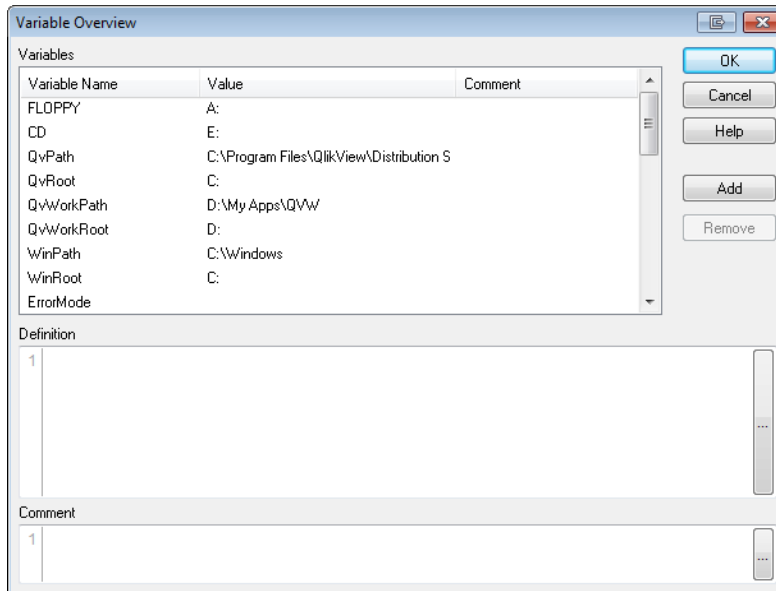
Enable this option if you want the alert message to be shown as a pop-up balloon when the alert is fired.

- Send mail** Enable this option 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.



# 18 Variable Overview



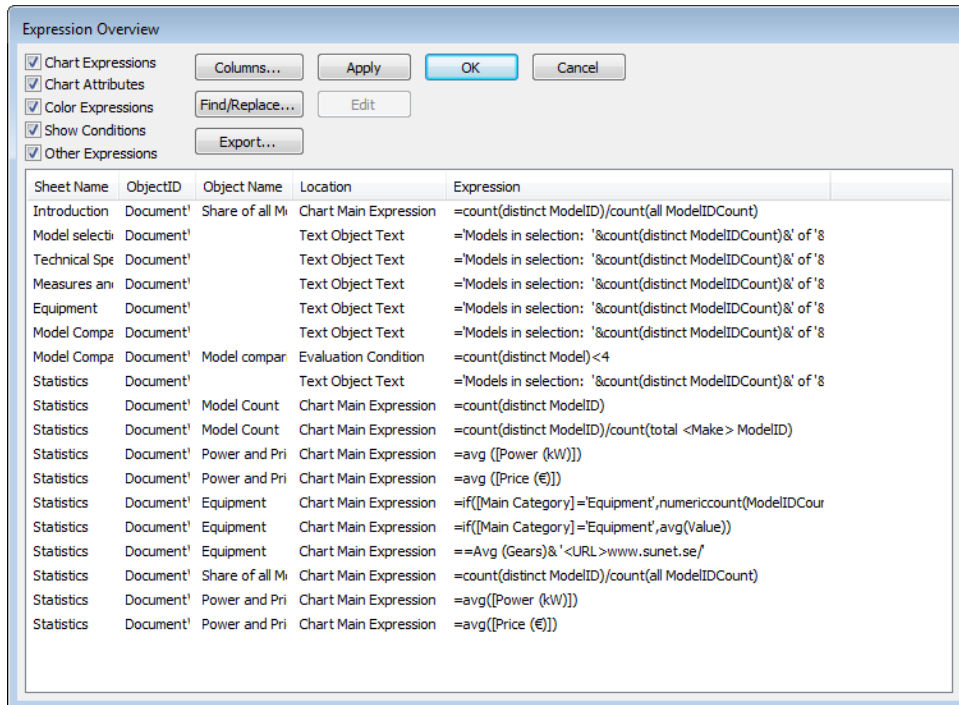
### *The Variable Overview dialog*

The **Variable Overview** dialog displays all non-hidden variables and their values in a single list. Together with the *Dollar-Sign Expansions* (page 290) feature in QlikView expressions, the variable overview may be used as a crude expression repository.

- Variables**                      In this two-column list the names of all non-hidden variables and their respective values are displayed.
  
- Definition**                    Highlight a variable in the list above to display its definition. The text may be freely edited. The changes will immediately be reflected in the **Value** column of the list above. Click on the ... button to open the **Edit Expression** dialog.
  
- Comment**                        A commentary field where the creator of a variable can describe the purpose and function of a variable.
  
- Add**                             Opens the **New Variable** dialog, where a new variable can be defined.
  
- Remove**                         Highlight a variable in the list and click this button to delete a variable.



# 19 Expression Overview



## The Expression Overview dialog

This dialog 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. Select one or more expressions from the list. A selected line appears in black. Click below the last expression to deselect it.

In the top left hand corner of the dialog you the following check boxes can be enabled. Select more than one check box.

- Chart Expressions
- Chart Attributes
- Color Expressions
- Show Conditions
- Other Expressions

### Columns...

Opens the **Columns** dialog where you may customize which columns should appear in the **Expression Overview** dialog. The **Expression** column cannot be turned off.

### Find/Replace...

Opens the **Find/Replace** dialog where you may search for and replace specific text within the current list of expressions. It is also possible to make multiple or bulk changes to the expressions.

### Export...

Use this button to export the table below 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.

**Edit...** After selecting a single expression in the list you may use this button to open the **Edit Expression** dialog for the selected expression.

The expression list in the lower part of the dialog may contain the following columns, based on settings in the **Columns** dialog (see below) :

<b>Sheet ID</b>	The sheet ID of the sheet where the expression occurs. Empty for document expressions.
<b>Sheet Name</b>	The name (tab title) of the sheet where the expression occurs. Empty for document expressions.
<b>Object ID</b>	The sheet object ID of the sheet object where the expression occurs. Empty for document and sheet expressions.
<b>Object Name</b>	The name of the object where the expression occurs.
<b>Location</b>	The location of the expression within properties with 'friendly name', i.e. a simple explanation of the type of expression.
<b>Label</b>	The label of the main expression of the object.
<b>Location (Full)</b>	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.
<b>Expression</b>	The expression definition.

Changes made from this dialog can be undone by the regular **Undo** command.

## 20 Internal Files

### 20.1 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 in *Internal Files in QlikView Documents (page 157)*.

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 *Internal Files in the QlikView Program File (page 157)*.

### 20.2 Internal Files in QlikView Documents

The **info** prefix to *Load (page 253)* and *Select (SQL) (page 270)* statements in the script 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://fieldname/fieldvalue |
qmem://fieldname<index>
```

where *index* is the internal index of a value within a field.

#### Examples:

```
'qmem://Country/Austria'
'qmem://MyField/34'
```

### 20.3 Internal Files in the QlikView Program File

A set of image files and sound files are already embedded in the QlikView program file. These may be referenced directly from the layout, without the need for any special preparations. The syntax for reference is:

```
qmem://<builtin>/filename
```

where *filename* is one of the filenames listed below (including extension).

#### Examples:

```
'qmem://<builtin>/Arrow_N_G.png'
='qmem://<builtin>/Smiley'&if(sum(Result)<0,1,3)&'_Y.png'
```

## Arrow Images

The arrows set permits you to display arrows in eight directions and four colors. Filenames are coded as follows: The word **Arrow** followed by a map direction (**\_N**, **\_NE**, **\_E**, **\_SE**, **\_S**, **\_SW**, **\_W** or **\_NW**). This may be followed by **\_G** for green, **\_R** for red and **\_Y** for yellow. If no color is stated in the filename, a gray arrow is referenced.

## Other Images

A number of other images are also included in the set. These are usually available in several colors. Below is a list of the available images.

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	Question.png
Question_G.png	Question_R.png	Question_Y.png	Exclamation.png
Exclamation_G.png	Exclamation_R.png	Exclamation_Y.png	

## 20.4 Sounds

A number of sound files (.wav) are also included as bundled resources within QlikView, 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_script.wav
qmem://<bundled>/sounds/qv_unlock.wav
qmem://<bundled>/sounds/qv_undo.wav
```

---

**Note!**

This list may change in future versions of QlikView.

---

---

# Part 4 Script

## 21 Introduction to Loading Data

This is a very brief description of the methods by which source data is accessed and used by QlikView. This introduction has been inserted in order to direct less experienced users to other topics that may otherwise be difficult to identify right away.

A very powerful aspect of QlikView is its ability to combine data from numerous data sources into a single QlikView document. This includes the possibility to read from several databases if needed. The program can also combine data from Internet sources and other servers or files.

QlikView runs a script which is managed in the script editor (*Edit Script Dialog (page 175)*), in order to connect to and retrieve data from various data sources. In the script the fields and tables to load are specified. It is also possible to manipulate the data structure by using special script statements and expressions. The script uses a special *Script Syntax (page 219)*.

The script is executed by clicking the **Reload** button. After the script execution, the **Select fields** dialog opens. There you choose the *Fields (page 171)* to display in list boxes on the sheet in QlikView.

During script execution, QlikView identifies common fields from different tables (key fields) in order to associate the data. The data structure of the data in the QlikView document can be monitored in the *Table Viewer (page 183)*, accessed from the **File** menu. Changes to the data structure can be achieved by *Renaming Fields (page 384)* in order to obtain different associations between tables.

After the data has been loaded into QlikView, it is stored in the QlikView document (*QlikView File (page 45)*). This is the heart of the program's functionality and it is characterized by the unrestricted manner in which data is associated, by its vast number of possible dimensions, by its speed of analysis and by its compact size. The data is held in the RAM when the QlikView document is open.

Finally, it should be noted that due to the role and nature of the QlikView database, the analysis in QlikView is always conducted while the document is off-line to its data sources. Consequentially, in order to refresh the data, the script has to be reloaded.

### 21.1 Loading Data from Files

QlikView can read data from files representing a table, in which the fields are separated by a delimiter like commas, tabs or semicolons. Other possible formats are dif files (Data Interchange Format), fix files (fixed record length), HTML tables, Excel files, xml files and the native QVD (*QVD Files (page 409)*) and QVX files. In most cases, the first line in the file holds the field names.

Files are loaded by means of a *Load (page 253)* statement in the script.

Instead of typing the statements manually in the script editor, you can use the Table file wizard to generate the statements automatically. See *File Wizard: Source (page 197)* for more information.

In **load** statements, the full set of *Script Expressions (page 295)* may be used.

To read data from another QlikView document, you may use a *Binary (page 223)* statement.

For help on how to load data from databases see *Loading Data from Databases (page 160)*.

### 21.2 Loading Data from Databases

Data from commercial database systems is loaded into QlikView via the Microsoft OLE DB/ *ODBC* interface. To do so, you must install a *Driver* to support your DBMS and you must configure the database as ODBC data source.

When this is done, you can set up a connection to the database in the *Edit Script Dialog (page 175)* by clicking the **Connect** button. This generates a *Connect (page 228)* statement in the script.

After that, you define the fields and tables to be loaded in a *Select (SQL) (page 270)* statement. To create this statement, click the **Select** button. See *Create Select Statement (page 189)* for more information.



You can also define the fields and tables to be loaded with a **Direct Query** statement. See *Direct Discovery (page 161)* for more information.

You can also load aggregated query results from large datasets without loading them into the QlikView data model, using the *Direct Discovery (page 161)* function.

## 21.3 Direct Discovery

Direct Discovery expands the associative capabilities of the QlikView in-memory data model by providing access to additional source data through an aggregated query that seamlessly associates larger data sets with in-memory data. Direct Discovery enhances business users' ability to conduct associative analysis on big data sources without limitations. Selections can be made on in-memory and Direct Discovery data to see associations across the data sets with the same QlikView association colors--green, gray, and white. Charts can analyze data from both data sets together.

Data is selected for Direct Discovery using a special script syntax, "DIRECT QUERY." Once the Direct Discovery structure is established, Direct Discovery fields can be used along with in-memory data to create QlikView objects. When a Direct Discovery field is used in a QlikView object, an SQL query is run automatically on the external data source. See *Differences between Direct Discovery and In-Memory Data (page 165)*.

While a QlikView application can access multiple tables, only one database table or one view can be accessed with Direct Discovery.

QlikView Server and Publisher require certain settings with running applications that use Direct Discovery. See *Direct Discovery with QlikView Server and Publisher (page 169)*.

### Types of Direct Discovery Fields

Within Direct Discovery, there are three types of data fields: DIMENSION, MEASURE, and DETAIL. The types are set on data fields when the Direct Discovery selection is made using the **Direct Query** statement in the load script. DIMENSION fields are loaded in memory and can be used to create associations between in-memory data and the data in Direct Discovery fields.

MEASURE fields, on the other hand, are recognized on a "meta level." MEASURE fields are not loaded in memory (they do not appear in the Table View). The purpose is to allow aggregations of the data in MEASURE fields to take place in the database rather than in memory. Nevertheless, MEASURE fields can be used in QlikView expressions without altering the expression syntax. As a result, the use of Direct Discovery fields from the database is transparent to the end user.

---

Note:

Due to the interactive and SQL-specific nature of Direct Discovery, only certain QlikView objects can use MEASURE fields. List boxes, Multi boxes, Statistics boxes, and mini charts are not supported with MEASURE fields.

---

DIMENSION fields are used to create associations between in-memory data and Direct Discovery data. Also, unlike MEASURE fields, DIMENSION fields can be used to define List boxes when you want to make Direct Discovery data available for selection in the QlikView user interface. Direct Discovery DIMENSION fields are also used to define dimension values in charts.

---

Note:

When DIMENSION fields are used in List boxes, the data values displayed in the List boxes are not updated with direct queries. The Direct Discovery fields' data is updated only by reloading.

---

Typically, fields with discrete values that will be used as dimensions should be loaded with the DIMENSION keyword, whereas numeric data that will be used in aggregations only should be marked as MEASURE fields. DIMENSION fields cannot be used in object expressions.

DETAIL fields provide information or details, like "Comment" fields, that a user may want to display in a drill-to-details Table Box. DETAIL fields cannot be used in chart expressions; they cannot be used in any aggregations. Nor can they be used in List Boxes. Fields designated as DETAIL commonly contain data that cannot be aggregated in any meaningful way, like "Comments."

Nevertheless, any field can be designated as a DETAIL field. Designating a field as DETAIL indicates that it will not be used in any aggregations, that it will only appear in Table Boxes.

DETACH fields act like DIMENSION fields except that association queries are not formulated when a DETACH is selected in a List Box or Chart (see *Differences between Direct Discovery and In-Memory Data* (page 165)).

The following *Aggregation Functions* (page 298) can be used with MEASURE fields:

- Sum
- Avg
- Count
- Min
- Max

The following table summarizes the characteristics and usage of the Direct Discovery field types:

Field Type	In Memory?	Forms Association?	Used in List Box?	Used in Chart expressions?	Used in Table Box?
DIMENSION	Yes	Yes	Yes	Yes	Yes
MEASURE	No	No	In aggr() expressions only	Yes	Yes
DETAIL	No	No	No	No	Yes
DETACH	Yes	No	Yes	Yes	Yes

All Direct Discovery fields can be used in Table Boxes, and they can be used in combination with in-memory fields.

## Direct Discovery Fields in QlikView Objects

Once the Direct Discovery structure is setup, the Direct Discovery fields from the source table are available in the user interface. The fields specified as DIMENSION are available for selections and to enable associations between in-memory fields and Direct Discovery fields.

MEASURE fields are marked in the user interface.

The screenshot shows the 'Fields' dialog box in QlikView. It has four tabs: 'Fields', 'Functions', 'Variables', and 'Images'. The 'Fields' tab is selected. Inside the dialog, there are three dropdown menus: 'Aggregation' (empty), 'Table' (set to 'All Tables'), and 'Field' (set to 'SubTotal (Measure)'). To the right of these dropdowns are a percentage input field (set to '0'), a 'Show System Fields' checkbox (unchecked), and a 'Distinct' checkbox (unchecked). Below these is a 'Paste' button. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

As noted above, there are some restrictions on the use of MEASURE fields in QlikView objects. It is also important to know when you are using MEASURE fields because using them can slow down the user experience. Although the data values for the MEASURE fields are not loaded in memory, they consume memory and CPU when they are used in the user interface of the QlikView application. For example, big data sources may have a billion values in a field, and using such a field would have a significant impact on performance.

See *How To Create List Boxes and Table Boxes with Direct Discovery (page 499)* and *How To Create Charts with Direct Discovery (page 784)*.

## Supported Data Sources

QlikView Direct Discovery can be used against the following data sources, both with 32-bit and 64-bit connections;

- ODBC/OLEDB data sources - All ODBC/OLEDB sources are supported, including SQL Server, Teradata and Oracle.
- Custom connectors which support SQL – SAP SQL Connector, Custom QVX connectors for SQL compliant data stores.

Both 32-bit and 64-bit connections are supported.

For SAP, Direct Discovery can be used only with the QlikView SAP SQL Connector, and it requires the following parameters in SET variables:

```
SET DirectFieldColumnDelimiter=' ';
SET DirectIdentifierQuoteChar=' ';
```

SAP uses OpenSQL, which delimits columns with a space rather than a comma, so the above set statements cause a substitution to accommodate the difference between ANSI SQL and OpenSQL.

Direct Discovery can be used with Google Big Query and requires the following parameters in the set variables:

```
SET DirectDistinctSupport=false;
SET DirectIdentifierQuoteChar='[]';
SET DirectIdentifierQuoteStyle='big query'
```

Google Big Query does not support SELECT DISTINCT or quoted column/table names and has non-ANSI quoting configuration using '['].

## Supported Data Types

All data types are supported in Direct Discovery, though there may be cases in which specific source data formats need to be defined to QlikView. This can be done on the load script by using the "SET Direct...-Format" syntax. The following example demonstrates how to define the date format of the source database that is used as the source for Direct Discovery:

### Example:

```
SET DirectDateFormat='YYYY-MM-DD';
```

There are also two script variables for controlling how the Direct Discovery formats money-type values in the generated SQL statements

```
SET DirectMoneyFormat (default '#.0000')
SET DirectMoneyDecimalSep (default '.')
```

The syntax for these two variables is the same as for MoneyFormat and MoneyDecimalSep, but there are two important differences in the usage:

- This is not a display format, so it should not include currency symbols or thousands separators.
- The default values are not driven by the locale but are hard wired to the values. (Locale-specific formats include the currency symbol.)

See *Number Interpretation Variables* (page 366).

Direct Discovery can support the selection of extended Unicode data by using the SQL standard format for extended character string literals (N'<extended string>') as required by some databases, such as SQL Server. This syntax can be enabled for Direct Discovery with the script variable `DirectUnicodeStrings` (page 234). Setting this variable to "true" enables the use of "N" in front of the string literals.

## Security

The following security best practices should be taken into consideration when using Direct Discovery:

- All of the users using the same QlikView application with the Direct Discovery capability use the same connection. Authentication pass-through and credentials-per-user are not supported.
- Section Access is not supported.
- It is possible to execute custom SQL statements in the database with a NATIVE keyword expression, so the database connection set up in the load script should use an account that has read-only access to the database.
- Direct Discovery has no logging capability, but it is possible to use the ODBC tracing capability. See the section on *Logging* (page 164).
- It is possible to flood the database with requests from the client.
- It is possible to get detailed error messages from the QlikView Server log files.

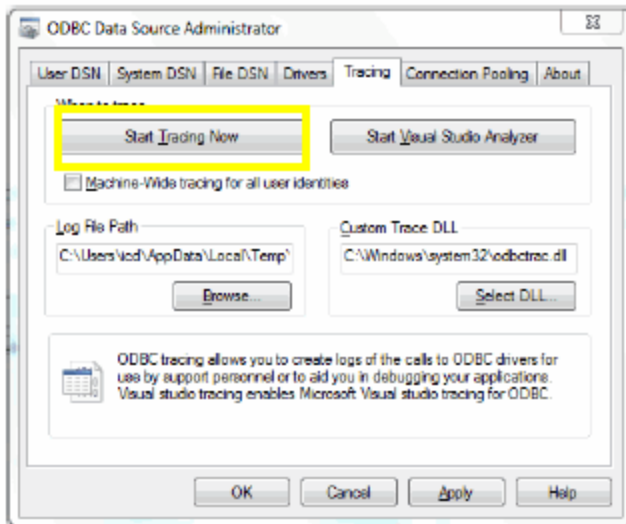
## Supported QlikView Functionality

Because of the interactive and SQL syntax-specific nature of Direct Discovery, several QlikView features are NOT supported:

- Advanced calculations (Set Analysis, complex expressions)
- Calculated dimensions
- Comparative Analysis (Alternate State) on the QlikView objects that use Direct Discovery fields
- Direct Discovery MEASURE and DETAIL fields are not supported on Global Search
- Binary load from a QlikView application with a Direct Discovery table
- Section access and data reduction
- Loop and Reduce
- Synthetic keys on the Direct Discovery table
- Table naming in script does not apply to the Direct table
- The use of wild card \* character after DIRECT QUERY keyword on the load script (DIRECT QUERY \*)
- Oracle database tables with LONG datatype columns are not supported.
- Big integers in scientific notation, outside range [-9007199254740990, 9007199254740991], can cause rounding errors and undefined behavior.

## Logging

Direct Discovery SQL statements passed to the data source can be recorded in the trace files of the database connection. For a standard ODBC connection, tracing is started with the ODBC Data Source Administrator:

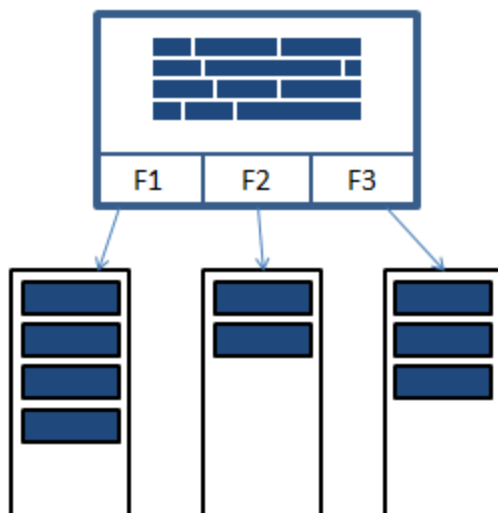


The resulting trace file details SQL statements generated through the user selections and interactions.

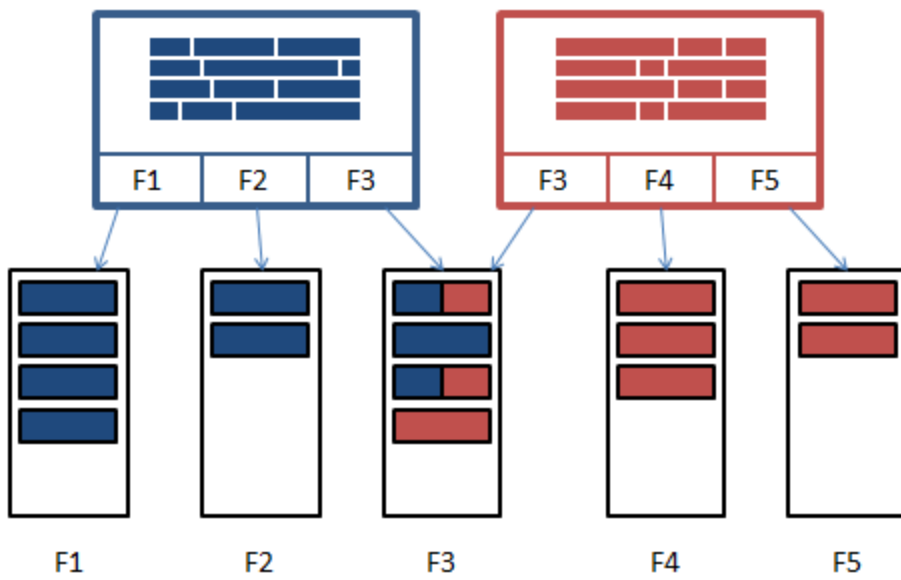
## 21.4 Differences between Direct Discovery and In-Memory Data

### The In-Memory Model

In QlikView's In-Memory Model, all unique values in the fields selected from a table in the load script are loaded into field structures, and the associative data is simultaneously loaded into the table. The field data and the associative data is all held in memory.

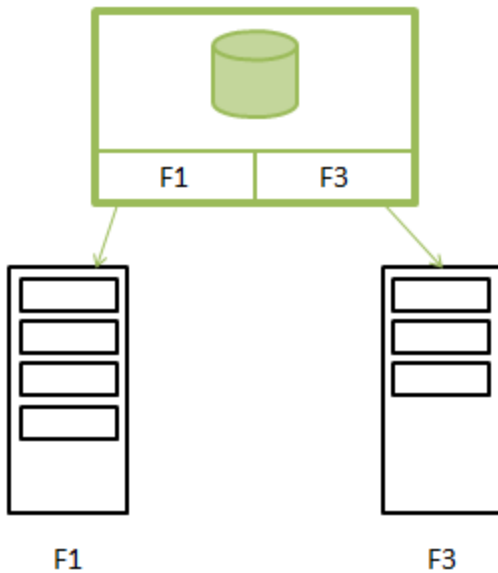


A second, related table loaded into memory would share a common field, and that table might add new unique values to the common field, or it might share existing values.

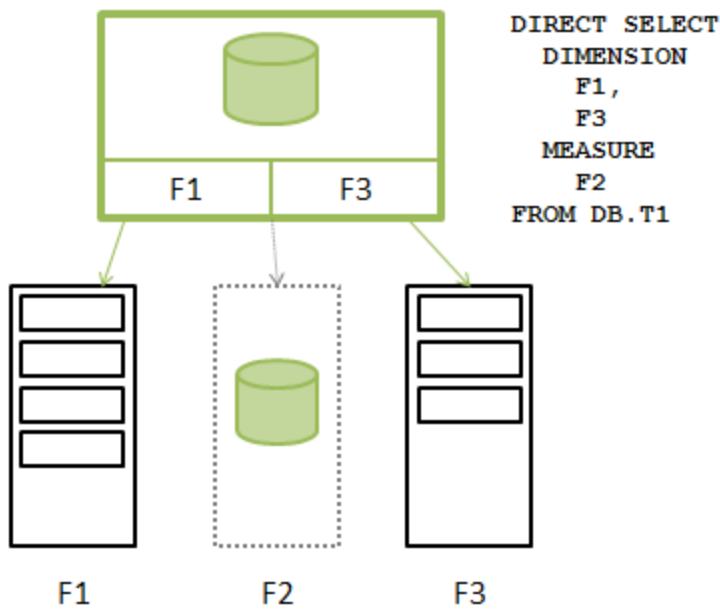


## Direct Discovery

When table fields are loaded with a Direct Discovery load statement (**Direct Query**), a similar table is created with only the DIMENSION fields. As with the In-memory fields, the unique values for the DIMENSION fields are loaded into memory. But the associations between the fields are left in the database.



MEASURE field values are also left in the database.



Once the Direct Discovery structure is established, the Direct Discovery fields can be used with certain QlikView objects (see *Direct Discovery Fields in QlikView Objects (page 162)*), and they can be used for associations with in-memory fields. When a Direct Discovery field is used in a QlikView object, QlikView automatically creates the appropriate SQL query to run on the external data. When selections are made, the associated data values of the Direct Discovery fields are used in the WHERE conditions of the database queries.

With each selection, the charts with Direct Discovery fields are recalculated, with the calculations taking place in the source database table by executing the SQL query created by QlikView. The calculation condition feature can be used to specify when charts should be recalculated. Until the condition is met, QlikView does not send queries to recalculate the charts.

## Performance Differences between In-Memory Fields and Direct Discovery Fields

In-memory processing is always faster than processing in source databases. Performance of Direct Discovery reflects the performance of the system running the database processing the Direct Discovery queries.

It is possible to use standard database and query tuning best practices for Direct Discovery. All of the performance tuning should be done on the source database. Direct Discovery does not provide support for query performance tuning from the QlikView application. It is possible, however, to make asynchronous, parallel calls to the database by using the connection pooling capability. The load script syntax to set up the pooling capability is:

```
SET DirectConnectionMax=10;
```

QlikView's caching also improves the overall user experience. See *Caching and Direct Discovery (page 169)* below.

Performance of Direct Discovery with DIMENSION fields can also be improved by detaching some of the fields from associations. This is done with the DETACH keyword on *Direct Query*. While detached fields are not queried for associations, they are still part of the filters, speeding up selection times.

While QlikView in-memory fields and Direct Discovery DIMENSION fields both hold all their data in memory, the manner in which they are loaded affects the speed of the loads into memory. QlikView in-memory

fields keep only one copy of a field value when there are multiple instances of the same value. However, all field data is loaded, and then the duplicate data is sorted out.

DIMENSION fields also store only one copy of a field value, but the duplicate values are sorted out in the database before they are loaded into memory. When you are dealing with large amounts of data, as you usually are when using Direct Discovery, the data is loaded much faster as a DIRECT QUERY load than it would be through the SQL SELECT load used for in-memory fields.

## Differences between Data In-Memory and Database Data

QlikView is case-sensitive when making associations with in-memory data. Direct Discovery selects data from source databases according to the case-sensitivity of the database. If a database is not case-sensitive, a Direct Discovery query might return data that an in-memory query would not. For example, if the following data exists in a database that is not case-sensitive, a Direct Discovery query of the value "Red" would return all four rows.

ColumnA	ColumnB
red	one
Red	two
rED	three
RED	four

An in-memory selection of "Red," on the other hand, would return only:

Red two

QlikView normalizes data to an extent that produces matches on selected data that databases would not match. As a result, an in-memory query may produce more matching values than a Direct Discovery query. For example, in the following table, the values for the number "1" vary by the location of spaces around them:

ColumnA	ColumnB
' 1'	space_before
'1'	no_space
'1 '	space_after
'2'	two

If you select "1" in a List Box for ColumnA, where the data is in standard QlikView in-memory, QlikView associates the first three rows:

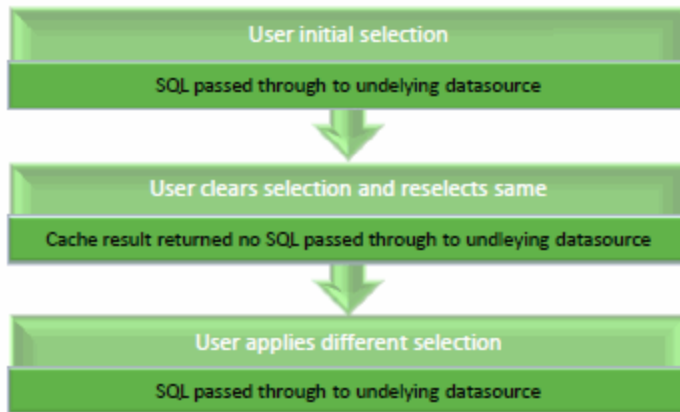
' 1'	space_before
'1'	no_space
'1 '	space_after



If the List Box contains Direct Discovery data, the selection of "1" might associate only "no\_space." The matches returned for Direct Discovery data depends on the database. Some return only "no\_space" and some, like SQL Server, return "no\_space" and "space\_after."

## Caching and Direct Discovery

QlikView's caching stores selection states of queries in memory. As the same types of selections are made, QlikView leverages the query from the cache instead of querying the source data. When a different selection is made, an SQL query is made on the data source. The cached results are shared across users.



A time limit can be set on caching with the *DirectCacheSeconds* (page 234) system variable. Once the time limit is reached, QlikView Server clears the cache for the Direct Discovery query results that were generated for the previous selections. QlikView then queries the source data for the selections and recreates the cache for the designated time limit.

The default cache time for Direct Discovery query results is 30 minutes unless the *DirectCacheSeconds* system variable is used.

## 21.5 Direct Discovery with QlikView Server and Publisher

### QlikView Server Settings

Some settings on QlikView Server should be reviewed if Direct Discovery is used on a QlikView application.

Note!

These settings will affect all QlikView documents that exist on the same QlikView Server.

#### Object Calculation Time Limit

As the Direct Discovery feature queries an external system from QlikView, the chart calculation time is dependent on the performance of the underlying system. It is advisable to set the **Object Calculation Time Limit** setting on QlikView Management Console high enough to allow enough time for the QlikView chart to get the direct discovery query results back from the data source. This setting is located on the **Performance** tab of QlikView Server listed on the QlikView Management Console.

#### Max Symbols in Charts

The Max Symbols in Charts setting is used to set the number of data points to be displayed on QlikView charts. Please note that as a Direct Discovery query can return many distinct values, it is advisable to review this setting to allow QlikView to display the desired number of data points on charts.

**QVS  
Time-Out  
Setting in  
the Con-  
fig.xml  
File**

When Direct Discovery is used to query an external system separate from the QlikView Server, the Server time-out setting in the config.xml file should be adjusted to allow enough time for QlikView to get the query results back. Failure to allow enough time to complete queries results in a "Lost connection to server" error when using the Ajax client.

The default setting is 60 seconds. The setting may need to be increased to the maximum query time. The config.xml file is located in the C:\ProgramData\QlikTech\WebServer directory. The XML tag is <QvsTimeout>. Note that during upgrades, the setting will be overwritten with the default value.

## QlikView Publisher

QlikView Publisher requires read access to database tables used by Direct Discovery. When QlikView applications that employ Publisher use Direct Discovery, ensure that the service account running Publisher has the required read access. If it does not, Publisher will not be able to read the Direct Discovery table during scheduled data refreshes.

## 22 Variables and Fields

This chapter explains the different data carrying entities in QlikView, most importantly **Fields** and **Variables**.

### 22.1 Fields

Fields are the primary data-carrying entities in QlikView. A field typically contains a number of values, called field values. In database terminology we say that the data processed by QlikView comes from data files. A file is composed of several *fields* where each data entry is a *record*. The terms *file*, *field* and *record* are equivalent to *table*, *column* and *row* respectively. The QlikView AQL logic works only on the fields and their field values.

Field data is retrieved by script via *Load* (page 253), *Select (SQL)* (page 270), *Direct Query* or *Binary* (page 223) statements. The only way of changing data in a field (apart from *Input Fields* (page 172)) is by re-executing the script. The actual field values can not be manipulated by the user from the layout or by means of Automation. Once read into QlikView they can only be viewed and used for logical selections and calculations.

Field values consist of numeric or alphanumeric (text) data. Numeric values actually have dual values, the numeric value and its current, formatted text representation. Only the latter is displayed in sheet objects etc. In the QlikView layout the content of a field is represented in a *List Box* (page 473).

### Field Tags

There are three different types of system tags, script generated system tags that cannot be altered by the user, script generated system tags that can be altered in the script and system tags that are set interactively by the user. System tags are always preceded by a \$ sign.

The following system tags are automatically generated at the end of script generation. These cannot be changed by the user:

- \$system - denotes a system field.
- \$key - denotes a key field.
- \$keypart - denotes that the field is part of one or more synthetic keys.
- \$synthetic - denotes a synthetic key.

The following tags are also automatically generated at the end of script generation, but may be altered or overridden using script syntax, see *Tag Field* (page 278) and *Untag Field* (page 280).

- \$hidden - denotes a hidden field.
- \$numeric - all (non-null) values in the field are numeric.
- \$integer - all (non-null) values in the field are integers.
- \$text - no values in the field are numeric.
- \$ascii - field values contain only standard ascii characters.
- \$date - all (non-null) values in the field can be interpreted as dates (integers).
- \$timestamp - all (non-null) values in the field can be interpreted as time stamps.

The following tags are set in the *Document Properties: Tables* (page 444) dialog. They can be enabled and disabled by the user:

- \$dimension - denotes a field recommended for use in chart dimensions, list boxes, etc.
- \$measure - denotes a field recommended for use in expressions.

The user can also add custom tags. These are added either in the script using *Script Syntax* (page 219) or in the *Document Properties: Tables* (page 444) dialog. These custom tags may not use the same name as any system tag.

**Note!**

If the same tags are manipulated in both the script and in *Document Properties: Tables (page 444)* the settings in the script will prevail.

---

## 22.2 Input Fields

Previously it was not possible to alter data in *Fields (page 171)* without running the script. However, 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.

---

**Note!**

Input fields are not meant for large amounts of data, as they store data significantly less efficiently than regular fields.

---

When using input fields, 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 (page 246)* 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 it is possible to see an input icon. Clicking on 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!**

A calculated field cannot be used as input field. The input field functionality will automatically be disabled.

---

A table chart expression cell must contain a *Special Input Field Aggregation Functions (page 817)* to 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".

---

**Note!**

If a file is loaded using the **binary** statement, fields defined as input fields will not be treated as input fields.

---

Version 9 introduced an extension to the input fields, namely the possibility to specify relative change. The following syntax applies (*n* is a number):

<b>%+n</b>	increases the current value by n%
<b>%-n</b>	decreases the current value by n%
<b>+=n</b>	increases the current value by n
<b>-=n</b>	decreases the current value by n
<b>*=n</b>	multiplies the current value with n
<b>/=n</b>	divides the current value with n

**Examples:**

- %+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.

## 22.3 Variable

A variable in QlikView is a named entity, containing a single data value. A variable typically acquires its value from a **Let**, **Set** or other control statement, via an automation call or via an *Input Box* (page 539) 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 *equals* 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.

## 22.4 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.

In general 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 as 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 **Document Properties: Groups** page.

### 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.

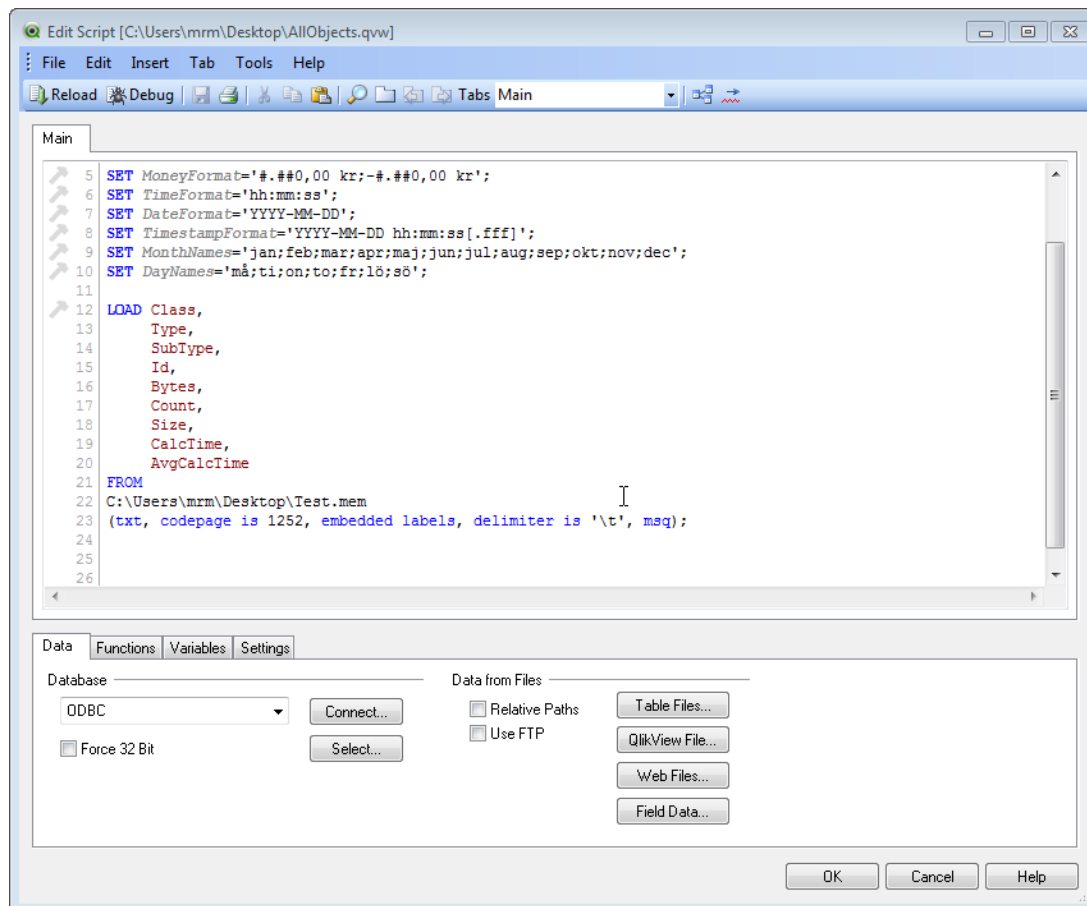
For more information about script expressions see *Script Expressions* (page 295).

For more information about chart expressions see chapter *Chart Expressions* (page 785).


For more information about the use of expressions in other layout objects see chapter *Calculated Formulas* (page 871).



## 23 Edit Script Dialog



*The Edit Script dialog*

The **Edit Script** dialog is opened from the *File Menu* (page 47) or by clicking on the **Edit Script** symbol  in the toolbar.

Here it is possible to enter and execute a script that connects the document 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* (page 219) and *Script Expressions* (page 295).

The **Edit Script** dialog uses autocomplete, so when typing, the program predicts what is wanted to type without having to type it completely. The predictions include words that are part of the script syntax. The script is also color coded by syntax components. It may be customized by choosing **Tools** and **Editor Preferences**.

At the top of the dialog a menu bar, with various script related commands, is found. The most frequently used commands also appear in the toolbar. In the toolbar there is also a dropdown list for the tabs of the script.





### Menus in the Edit Script Dialog

#### File Menu

<b>Reload</b>	Executes the script, closes the dialog and opens the <b>Fields</b> page in the <b>Sheet Properties</b> dialog. This command is also available on the <b>Edit Script</b> dialog toolbar.	
<b>Debug...</b>	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. This command is also available on the <b>Edit Script</b> dialog toolbar.	
<b>Edit Hidden Script...</b>	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.	
<b>Change Hidden Script Password...</b>	Opens the <b>New Hidden Script Password</b> dialog where a new password for access to the hidden script can be set.	
<b>Create Hidden Script</b>	Opens the <b>New Hidden Script Password</b> 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.	
<b>Remove Hidden Script</b>	Removes the hidden script from the <b>Edit Script</b> dialog.	
<b>Save Entire Document</b>	Saves the active document in a file. Data, script and layout are saved. This command can also be invoked by the following keyboard shortcut: Ctrl+S. This command is also available on the <b>Edit Script</b> dialog toolbar.	
<b>Save Entire Document As...</b>	Saves the active document in a new document file under a new name.	
<b>Open External Script File</b>	Makes it possible to browse for a file containing a script and opens it in a new tab in the script pane, where it can be edited with the help of QlikView's color coding.	
<b>Save External File</b>	Saves the external script file.	
<b>Export to Script File...</b>	Saves the entire script in a text file to be specified in the <b>Save script</b> dialog that appears. The file will have the extension .qvs.	
<b>Print Tab...</b>	Allows the printing of the active script tab on an optional printer. This command can also be invoked by the following keyboard shortcut: Ctrl+P. This command is also available on the <b>Edit Script</b> dialog toolbar.	
<b>Print All Tabs...</b>	Allows the printing of the entire script on an optional printer. Hidden script will not be printed via this command.	
<b>Table Viewer</b>	Opens the <i>Table Viewer</i> (page 183) 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 by the following keyboard shortcut: Ctrl+T.	

## Edit Menu



<b>Undo</b>	Undoes the latest change (multiple step undo is possible). Equivalent to pressing Ctrl+Z.	
<b>Redo</b>	Redoes the latest <b>Undo</b> . Equivalent to pressing Ctrl+Y.	
<b>Cut</b>	Exports the selected text to the Clipboard. This command is also available on the Edit Script dialog toolbar. Equivalent to pressing Ctrl+X.	
<b>Copy</b>	Copies the selected text to the Clipboard. This command is also available on the <b>Edit Script</b> dialog toolbar. Equivalent to pressing Ctrl+C.	
<b>Paste</b>	Pastes the contents of the Clipboard into the dialog at the position of the cursor. This command is also available on the <b>Edit Script</b> dialog toolbar. Equivalent to pressing Ctrl+V.	
<b>Delete</b>	Deletes the selected script text. Equivalent to pressing Del.	
<b>Clear Entire Script</b>	Clears the script, i.e. removes all the text except the initial autogenerated <b>set</b> statements.	
<b>Select All</b>	Selects the entire script text. Equivalent to pressing Ctrl+A.	
<b>Find/Replace...</b>	Opens a dialog which makes it possible to find and replace numbers or characters in the script. This command is also available on the <b>Edit Script</b> dialog toolbar. Equivalent to pressing Ctrl+F.	
<b>Goto...</b>	Opens a dialog which makes it possible to go to a specified line number in the script. Equivalent to pressing Ctrl+G.	
<b>Upper Case</b>	Puts the selected script text in upper case.	
<b>Lower Case</b>	Puts the selected script text in lower case.	
<b>Invert Case</b>	Inverts the casing of the selected script text.	
<b>Capitalize</b>	Capitalizes the selected script text.	
<b>Comment</b>	Transforms a text row in the script to a comment. See also: <i>Rem (page 265)</i>	
<b>Uncomment</b>	Transforms the text row back into original script text.	
<b>Indent</b>	Indents the highlighted lines in the script.	
<b>Outdent</b>	Outdents the highlighted lines in the script.	

## Insert Menu

<b>Set Statement</b>	Opens the <i>Set Statement Wizard (page 181)</i> where it is possible to enter a new set statement and choose a predefined statement.
<b>Environment Variables</b>	Inserts the number interpretation variables in the script based on the operating system defaults.
<b>Script File</b>	Makes it possible to 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.

<b>Include Statement</b>	Opens the <b>Include Script Files</b> dialog, where it is possible to browse for a script file with one of the following formats: qvs (a previously saved Qlik-View script file), txt or sql.
<b>Domain SID</b>	Retrieves the NT domain security ID for use in the script. The NTDOMAINSID is employed in section access as one of the reserved fields that govern NT security.
<b>Test Script</b>	Inserts an autogenerated test script.
<b>Load Statement</b>	Inserts a load statement, either <b>From File</b> or <b>Inline</b> . Choosing <b>Inline</b> opens the <i>Inline Data Wizard (page 194)</i> with the help of which it is possible to create load inline statements from a spreadsheet style control.
<b>Section Access</b>	Inserts a section access statement either from a Publisher Authorization table or Inline. Choosing Publisher Authorization opens the <i>File Wizard: Options (page 209)</i> . Choosing Inline opens the <i>Access Restriction Table Wizard (page 195)</i> .
<b>Connect Statement</b>	Inserts a connect statement in the script.
<b>Disconnect Statement</b>	Inserts a disconnect statement in the script.

## Tab Menu

The **Tab** menu helps to organize the structure of the script.

<b>Add Tab...</b>	Adds a new script tab. The script will be executed tab by tab in order from left to right. If the current script tab is part of the hidden script, the new tab will also be created within the hidden script.
<b>Insert Tab at Cursor...</b>	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.
<b>Rename...</b>	Opens a dialog for renaming the active tab.
<b>Promote</b>	Moves the active tab one step to the left. Tabs cannot be promoted to the left of the hidden script tab.
<b>Demote</b>	Moves the active tab one step to the right.
<b>Merge with Previous...</b>	All text on the active tab is moved to the end of the preceding tab and the active tab is deleted.
<b>Remove...</b>	Removes the active tab. The last remaining script tab cannot be removed.

## Tools Menu

<b>ODBC Administrator 64 bit...</b>	Opens the ODBC Data Source Administrator for 64-bit ODBC drivers.
<b>ODBC Administrator 32 bit...</b>	Opens the ODBC Data Source Administrator for 32-bit ODBC drivers.
<b>Editor Preferences</b>	Opens the <i>User Preferences: Editor (page 74)</i> page where it is possible to set the font and color of the different text types appearing in the script.

**Syntax Check** Checks the syntax of your script and stops at the first error.

## Help Menu

**Help** opens the html help for QlikView.

## Panes in the Edit Script Dialog

There are two panes in the **Edit Script** dialog: the script pane at the top, and the tool pane at the bottom of the dialog.

### Script Pane

The script pane contains the actual script. Each script line is numbered. The script may be divided into several parts appearing on separate tabbed pages which are executed from left to right.

If a hidden script is used, it may be viewed under a separate tab to the very left (provided that the password has been given).

The script is color coded by syntax components. The color coding may be customized by choosing **Editor Preferences** from the **Tools** menu.

### Tool Pane

The tool pane consists of four tabbed pages containing functions for script generation.

### Data Page

The **Data** page contains basic commands for getting data into QlikView:

The commands in the **Database** group are used to create a connection to and select fields from a data source. If a commercial DBMS is used it is possible to use ODBC or OLE DB as an interface between QlikView and the database. An ODBC or OLEDB driver that supports the DBMS must be installed on the computer before it is possible to retrieve data over the ODBC/OLE DB interface. Once the appropriate ODBC/OLE DB driver is installed, the database must be configured as an ODBC data source.

<b>OLE DB</b>	The connection to the data source is established using OLE DB.
<b>ODBC</b>	The connection to the data source is established using ODBC.
<b>QVSAAdminDataProvider.dll</b>	This custom connector gives the possibility to connect to the QlikView Server and load information from the DMS and collaboration objects.
<b>Custom...</b>	Any custom data source will also appear in the drop-down box for selection. 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 OLE DB 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.
<b>Force 32 Bit</b>	Normally, QlikView 64-bit uses 64-bit providers. In case a data source with a 32-bit driver is to be used, check this option to force the ODBC/OLEDB connect statement to a 32-bit provider.

**Connect...** Opens the **Data Link Properties** dialog in which a Data source can be selected, or the *Connect to Data Source (page 188)* dialog where an ODBC data source can be selected.

**Select...** Opens the *Create Select Statement (page 189)* dialog.

The **Data from Files** group is where data retrieval from other data sources can be made.

**Relative Paths** If this setting is enabled, QlikView will use relative paths instead of absolute paths in the script. Relative paths are normally required when a document 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** dialogs.

**Use FTP** Enable this setting to select files from an FTP server, when clicking **Table Files**, **QlikView Files** or **Include**.

**Table files...** Opens the *Open Local Files (page 192)* dialog listing 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 to use, then click Connect). When this is done, select a text file.

**QlikView File...** Opens the **Open QlikView File** dialog. When a file is selected a **binary** statement is generated in the first line of the script. This statement loads the data from the QlikView file but the layout settings will not be used.

**Web Files...** Opens the *File Wizard: Source (page 197)* dialog where it is possible to enter a URL as a source for the database table.

**Field Data** Opens the **File Wizard: Source** dialog where it is possible to load the contents of an already loaded field.

## Variables Page

On the **Variables** page, the controls for pasting syntax relating to QlikView variables, are found.

**Paste** Pastes the selected function into the script.

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

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.

## Functions Page

The **Functions** page contains tools for navigating and pasting QlikView standard functions.

**Function Category** Select a category in the drop-down list to see the corresponding functions in the **Function Name** list below.

<b>Function Name</b>	This drop-down list contains standard QlikView script functions.
<b>Paste</b>	Select a function in the <b>Function Name</b> list and paste it into the script at the cursor position.

## Variables Page

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

<b>Paste</b>	Pastes the selected variable into the script.
<b>Show System Variables</b>	If this check box is marked, the list in the <b>Variables</b> drop down will include the system variables.

## Settings Page

The **Settings** page contains security settings for the QlikView script.

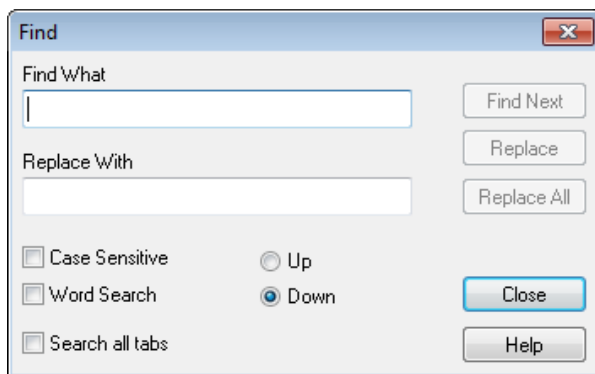
<b>Script Privileges</b>	<b>Open Databases in Read and Write Mode</b>
	If this option is selected the <b>mode is write</b> qualifier in select statements will be enabled for use.
	<b>Can Execute External Programs</b>
	If this option is selected execute statements may be used in the script.
<b>Settings</b>	<b>Scramble Connect User Credentials</b>
	If this check box is marked, the USERID and PASSWORD in <b>connect</b> statements will be scrambled in the script.

## 23.1 Set Statement Wizard

*The Set Statement wizard*

<b>Current Set Statement</b>	<p><b>Variable Name</b> Either enter a name of a new variable or edit the predefined variable chosen below. The predefined variable is inserted here when clicking <b>Paste</b>.</p> <p><b>Variable Value</b> If the name of a new variable is entered above, define the value here. If a predefined variable value has been chosen below it is possible to edit it here.</p>
<b>Predefined Set Statements</b>	<p><b>Variable Groups</b> Choose what kind of variable to use.</p> <p><b>Variables</b> Choose the variable to use.</p> <p><b>Predefined Values</b> Choose between the predefined values in the variable.</p> <p><b>Paste</b> Click the button to move the predefined values up to <b>Current Set Statement</b> for editing.</p>

## 23.2 Find/Replace (Script)



*The Find/Replace dialog*

This dialog is used for searching for specific text strings within the script and for making multiple or bulk changes to it. It opens from the **Edit** menu in the **Edit Script** dialog.

<b>Find What</b>	The text string to search for.
<b>Replace With</b>	The text to replace the search string with.
<b>Find Next</b>	Moves selection to the next occurrence of the search string.
<b>Replace</b>	Makes a replacement in the selected section.
<b>Replace All</b>	Makes replacements in all occurrences of the search string.
<b>Case Sensitive</b>	If this check box is marked, the text search will be case sensitive.
<b>Word Search</b>	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).
<b>Search All Tabs</b>	If this check box is marked, the find/replace operation will be made over all script tabs.
<b>Up</b>	Mark this radio button to search up through the script.

**Down** Mark this radio button to search down through the script.

## 23.3 Hidden Script

A hidden script is a special 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 accessing hidden script in a document (thereby creating one) the new password has to be confirmed. After this the Hidden Script tab will appear to the left of all other script tabs and remain there until the document is closed.

### Note!

If a hidden script is used, the **binary** command cannot be used in the normal script.

### Note!

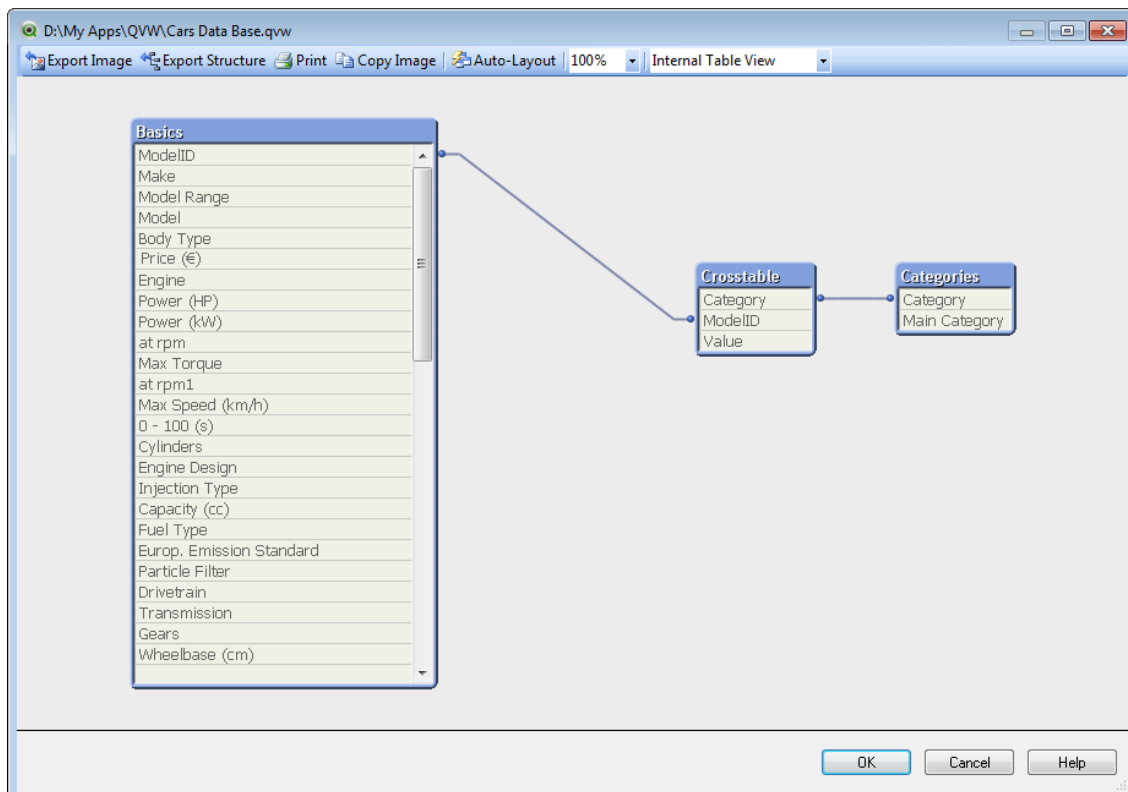
The **Progress** dialog will not be updated during the execution of a hidden script, unless **Show Progress for Hidden Script** is marked in **Document Properties: Security** page. No entries will be made in the log file, if used.

### 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.

## 23.4 Table Viewer

The **Table Viewer** dialog is opened when choosing **Table Viewer** from the **File** menu or press Ctrl+T.



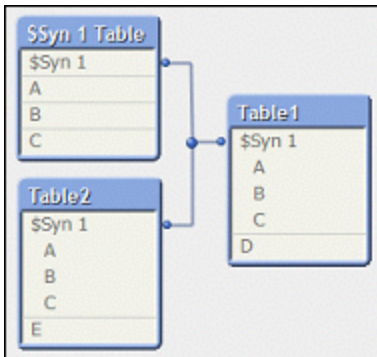
*The Table Viewer dialog*

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. Connector lines between the boxes show the associations. Where more than two lines meet there are connector points in the form of small dots.

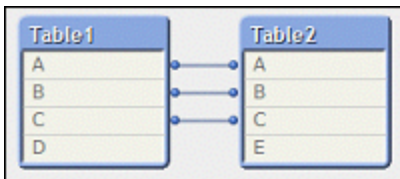
The tables, the connector points and also all points where the connectors bend can be moved by means of mouse drag and drop (see below). When re-arranging the layout like this, the changes will be saved with the document unless the dialog is exited with **Cancel**.

There are two views available, the layouts of which are stored separately. The two views are:

- The **Internal Table View** is the default view. It 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.



- The **Source Table View** shows data tables as QlikView reads them. Here there are 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 (that is other tables as well).
- **Table comments** are displayed in the tool tip.
- **Field comments** are displayed in the tool tip.
- **Tags** that are added in the script or in **Document Properties: Tables** page, including system tags, are displayed in the tool tip.

The following actions can be performed in the table layouts (click here to expand a list):






- |  |   |
|--|---|
| <b>Click on Table Caption</b>          | Highlights the table, all tables with direct logical associations with that table and the connections between them. |
| <b>Point at Table Caption and Drag</b> | Moves the table in the layout.  |
| <b>Hover Over Table Caption</b>        | Shows pop-up info about table.  |



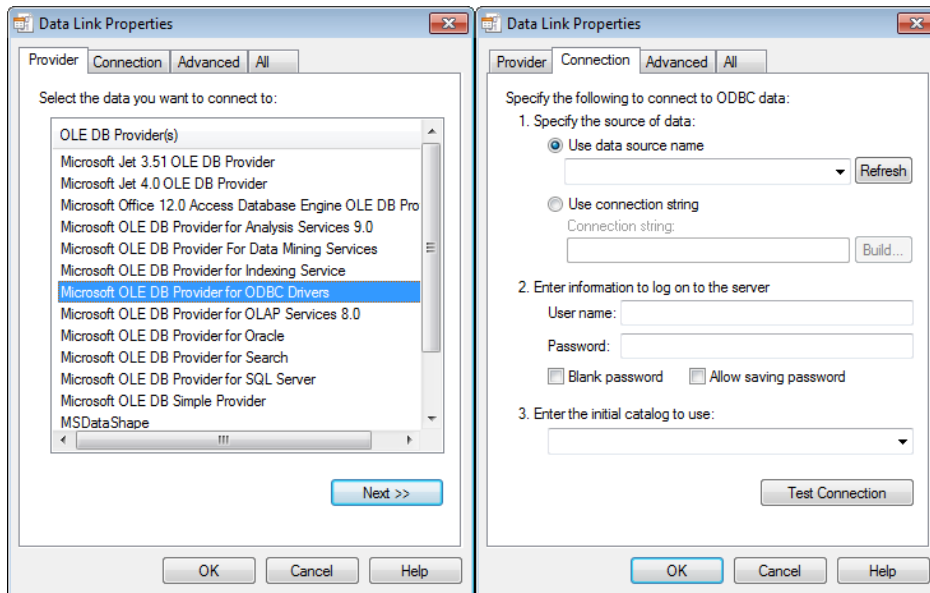
<b>Click on Field in Table</b>	Highlights the table, the field, all tables containing that field and the connections between them.
<b>Hover over Field in Table</b>	Shows pop-up info about field.
<b>Right-click in a Table</b>	Click on <b>Preview</b> to see a dialog of sample records from the loaded data.
<b>Point at Connector Point or Connector Bend and Drag</b>	Moves the connector point in the layout.
<b>Right-click on Connector Point or Connector Bend</b>	Moves the connector point back to automatic positioning.
<b>Click on Layout Background</b>	Cancels all highlighting of tables, fields and connectors.

Author	Title
Adams, Douglas	Hitchhikers Guide b
Adams, Douglas	Restaurant at the I
Adams, Douglas	So Long and Thank
Adams, Douglas	Titanic

The following commands are available in the dialog toolbar and buttons (click here to expand a list):

<b>Export Image</b>	Opens a file browser dialog making it possible to save the current table view as a bitmap or png image to a file on disc.	
<b>Export Structure</b>	By pressing this button it is possible to export the table structure of the document to a set of text files. These text files, one for the tables ( <i>filename.Tables.tab</i> ), one for the fields ( <i>filename.Fields.tab</i> ) and one for mapping in between ( <i>filename.Mappings.tab</i> ) can easily be read back into QlikView for further analysis with the full power of the QlikView logic. A dialog will appear where the target folder for the export can be chosen. The default is to put the files in the same folder as the QlikView document.	
<b>Print Image</b>	Opens the Print dialog from which it is possible to print the current table view. Equivalent to pressing Ctrl+P.	
<b>Copy Image</b>	Copies the current table view as an image to clipboard. Equivalent to pressing Ctrl+C.	
<b>Auto Layout</b>	Re-arranges the tables in the current view.	
<b>Zoom</b>	The zoom factor of the current view can be set in this drop-down box.	
<b>View</b>	Choose between <b>Internal Table View</b> and <b>Source Table View</b> (see above).	

## 23.5 Data Link Properties



### *Data Link Properties dialogs*

This dialog is used for creating a connection to an OLE DB data source, typically via ODBC. The dialog is opened when clicking the **Connect...** button in the **Data** page of the **Tool** Pane in the **Edit Script** dialog.

The first tab 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** tab.

On the **Connection** tab under **1**, **specify the source of data**, a previously configured data source is chosen. There are two different types of ODBC data sources that may come into question.

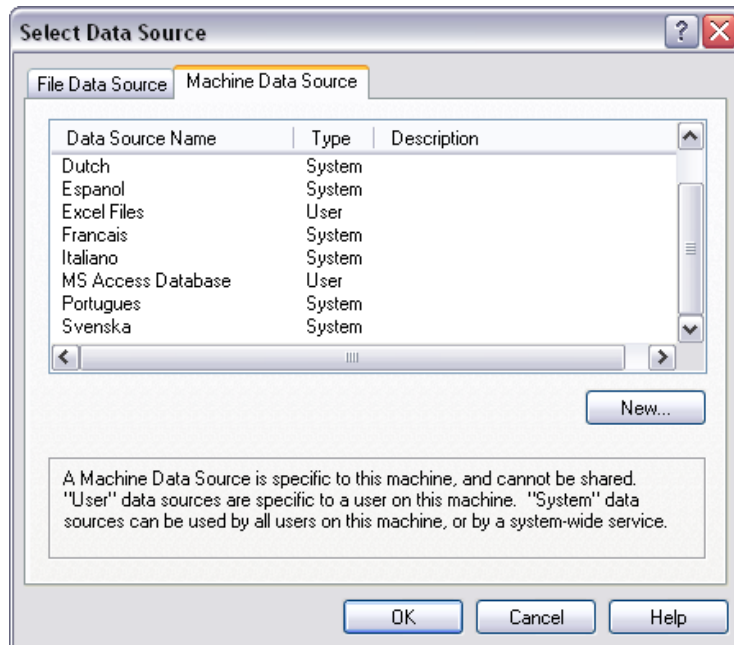
## Pre-Defined Data Sources

Data sources which have been predefined according to the procedure described in *Creating ODBC Data Sources (page 34)* 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 to use.

If necessary specify **User name** and **Password** for the data source. The connection can also be tested by pressing the **Test Connection** button.

Finally click the **OK** button. The connect statement is ready.

## Generic Data Sources



*Select Data Source dialog, Machine Data Source tab*

The Machine Data Source tab of the Select Data Source dialog

Generic data sources are used e.g. for database files. If a generic data source is to be used or a new source should be created, 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. Any of the two can be used. Each has a separate page in 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. 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.

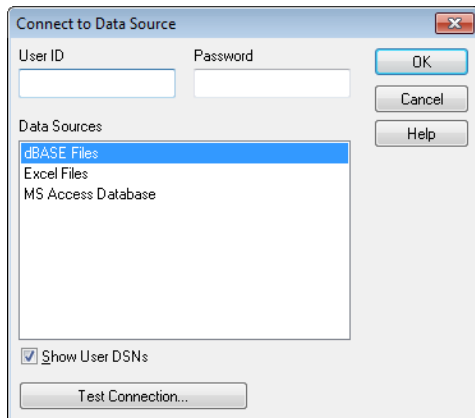
---

When hitting RETURN it returns to the second page of the **Data Link Properties** dialog.

Specify if necessary **User name** and **Password** for the data source. It is also possible to test the connection by pressing the **Test Connection** button.

Finally click the **OK** button. The **connect** statement is ready.

## 23.6 Connect to Data Source

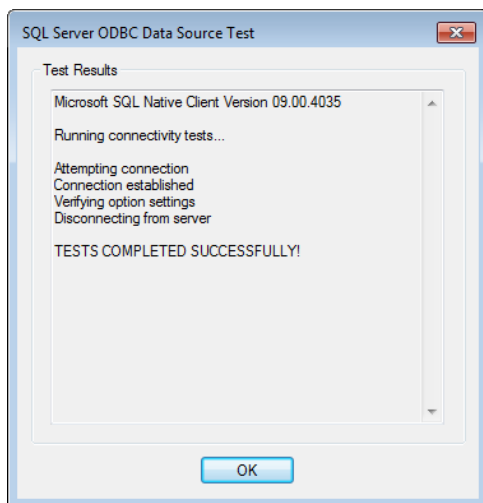


### *The Connect to Data Source dialog*

This dialog is used for creating a connection to an ODBC data source. The dialog is opened when clicking the **Connect** button in the **Data** page of the **Tool Pane** in the *Edit Script Dialog* (page 175) if the database option ODBC has been chosen.

On this page choose the data source. As default only the system DSNs are displayed. Check **Show User DSNs** to view all DSNs.

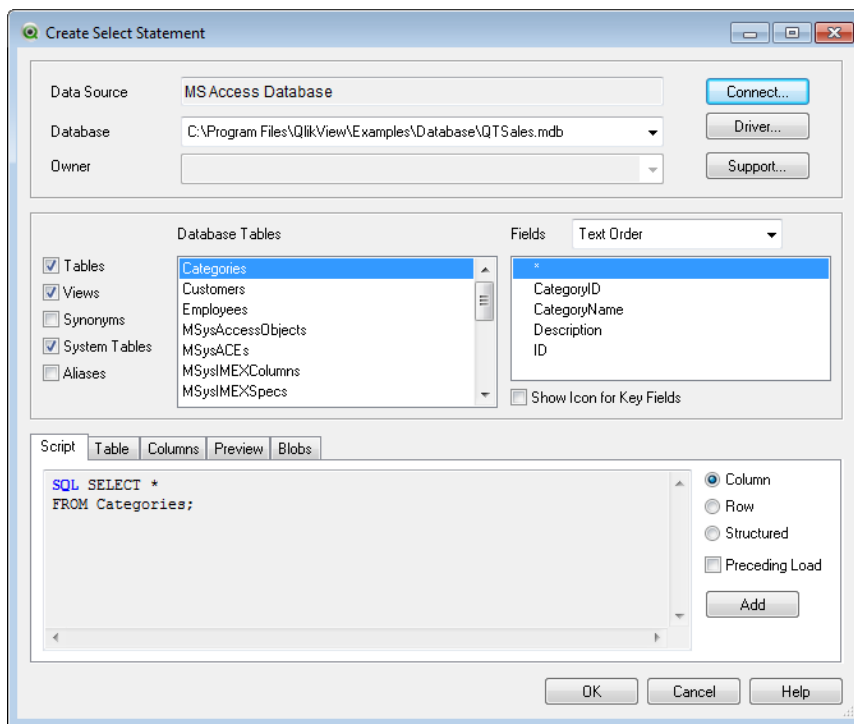
It is possible if necessary to specify **User name** and **Password** for the data source. The connection can also be tested by pressing the **Test Connection** button.



### *The Test Connection dialog*

Finally click the **OK** button. The **connect** statement is ready.

## 23.7 Create Select Statement

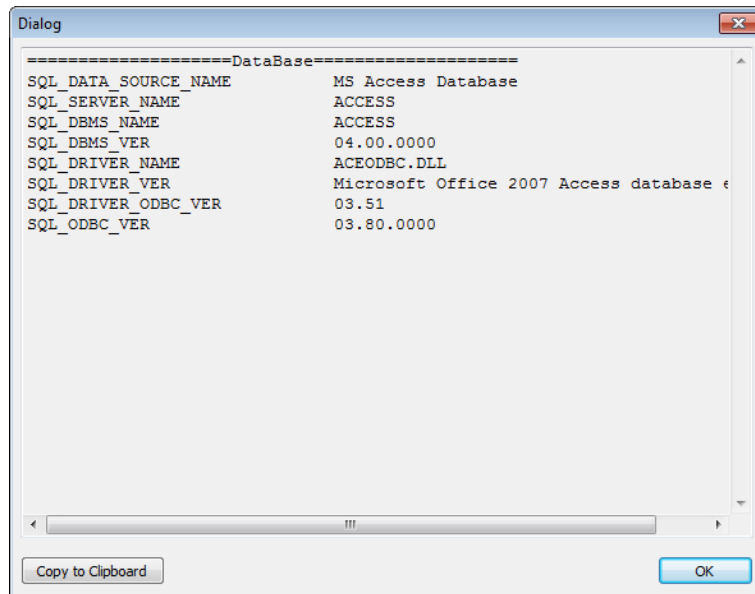


### *The Create Select Statement dialog*

This dialog is opened by clicking the **Select** button in the **Data** page of the tool pane in the *Edit Script Dialog* (page 175). It is used to define the tables and fields to be loaded from a previously selected data source.

The dialog consists of three groups. The first group contains information about the data source.

- |                    |   |
|--------------------|---|
| <b>Data Source</b> | Name of the current ODBC/OLE DB data source.  |
| <b>Database</b>    | It is possible to choose between available databases via the drop-down list.  |
| <b>Owner</b>       | The database owner is shown here. It is possible to choose between available owners via the drop-down list.   |
| <b>Connect</b>     | By clicking this button, the <i>Connect to Data Source</i> (page 188) dialog opens.   |
| <b>Driver</b>      | Click this button to open the <b>ODBC Driver Information</b> or <b>OLEDB Driver Information</b> dialog, containing information about supported functionality in the driver currently used.          |
| <b>Support</b>     | Opens a dialog with support information about the database. The information can be used whenever contacting QlikTech support and when reporting bugs or problems in relation to QlikView documents. |



#### *The Support dialog*

The second group is used for selecting tables and fields.

- |                                 |   |
|---------------------------------|---|
| <b>Database Tables</b>          | This list shows available database tables. Click on a table name to select it. The types of tables that appear in the list can be controlled via the check boxes to the left (see description below).   |
| <b>Fields</b>                   | This list shows all available fields in the selected table. Choose one or more fields from the list to be included in the <b>select</b> statement. Use " * " in order to select all of the field names at once. The selection of fields will be reflected in the <b>Script</b> page of the third group (see description below). Fields can be sorted in <b>Original Order</b> from the database or in alphabetical <b>Text Order</b> by selection in the drop-down control. |
| <b>Show Icon for Key Fields</b> | When this alternative is checked, any field that is defined as a key field in the source database will be shown with a key icon in the <b>Fields</b> list.  |
| <b>Tables</b>                   | This check box must be selected if regular database tables are to appear in the <b>Database Tables</b> list.  |
| <b>Views</b>                    | This check box must be selected if database views are to appear in the <b>Database Tables</b> list.   |
| <b>Synonyms</b>                 | This check box must be selected if database table synonyms are to appear in the <b>Database Tables</b> list.  |
| <b>System Tables</b>            | This check box must be selected if database system tables are to appear in the <b>Database Tables</b> list.   |
| <b>Aliases</b>                  | This check box must be selected if database table aliases are to appear in the <b>Database Tables</b> list.   |

The third group shows the generated **select** statement and information about the selected table and fields. The group contains the following pages:

## Script



The **Script** tab shows the script about to be generated for the **select** statement.

The three radio buttons to the right 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 option 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**                              Click this button to save the **select** statement generated so far. The generation of a new **select** statement is initiated 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** tab 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.

## Columns

The **Columns** tab shows detailed information about the selected columns (fields) of the current table.

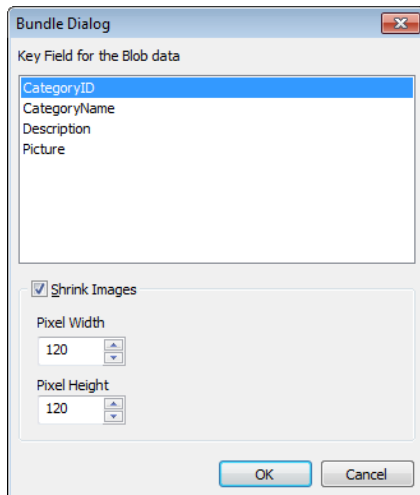
## Preview

The **Preview** tab shows a preview of the first rows that will be generated by the current **select** statement.

## Blobs

The **Blobs** tab makes it possible to bundle so-called Blobs (Binary Large Objects), in a QlikView document. It is usually a picture, a text file or similar, stored as a single entity in a database management system. The **i** 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.

The **Bundle...** button opens the **Bundle** dialog.



**Key Field for the Blob data** In the list select the blob that should be resized.

**Shrink Images** Enable the option to resize the blob. Apply **Pixel Width** and **Pixel Height** to resize the image to fit.

The setting does not handle aspect ratio.

---

#### Note!

It is only possible to bundle Blobs using ODBC.

---

Once the selections of table and fields has been made it is possible to click **OK** to insert the **select** statement(s) 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 the **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, click the **SQL** button.)

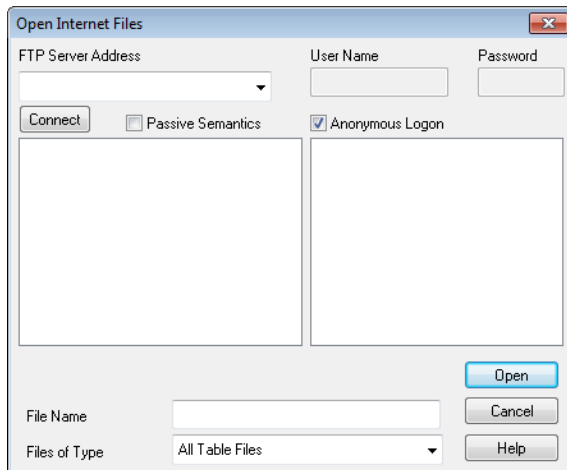
## 23.8 Open Local Files

This dialog is opened by the **Table files** button in the *Edit Script Dialog* (page 175).

In the **Open Local Files** dialog box it is possible to specify the *table files* that are to be loaded. For multiple file selections, use Ctrl-click or Shift-click. When the **OK** button is pressed, the selected file is interpreted in the **File Wizard**.



## 23.9 Open Internet Files or Open QlikView Document



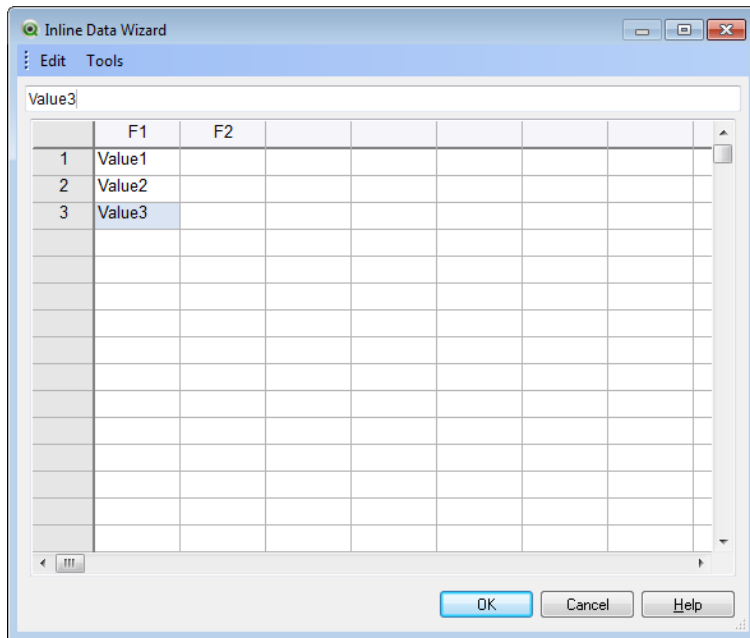
### *The Open Internet files dialog*

This dialog appears when clicking on 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 which button was used to open it, the dialog is called **Open QlikView Document** or **Open Internet Files**, but the functionality is the same. This dialog can also be used to open QlikView documents. This is done by choosing **Open FTP** from the **File** menu.

Select the server containing the file(s) that should be opened in the drop-down list box. If the server is not in the list (or if there is no list), enter a server name, then click **Connect**.

<b>FTP Server Address</b>	The first time connecting to the FTP server, the address has to be typed in here.
<b>User Name</b>	Unless <b>Anonymous Logon</b> is checked, the user name will be entered here.
<b>Password</b>	Unless <b>Anonymous Logon</b> is checked, the password will be entered here.
<b>Connect</b>	Click this button to connect to the selected server. Selecting a server directly from the list, bypasses this procedure.
<b>Passive Semantics</b>	To connect through a firewall, a passive FTP may have to be used.
<b>Anonymous Logon</b>	Lets the user log on without stating a user name and password.
<b>File Name</b>	The name(s) of the selected file(s) will appear here, when connected to the server.
<b>Files of Type</b>	Specify the desired file type here.
<b>Open</b>	Clicking this button generates a <b>load</b> statement that will enter into the load script in the <i>Edit Script Dialog</i> (page 175).

## 23.10 Inline Data Wizard



*The Inline Data Wizard*

The **Inline Data Wizard** dialog is opened from the **Insert** menu, **Load Statement**, **Load Inline**. It is used to create **load inline** statements 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 by means of an inline table. Each row is a record in the table. A data cell is selected by clicking it. A value may then be typed or pasted in from clipboard. Press Enter or an arrow key to accept the value and move to another cell.

The top (label) row is reserved for field labels. Double-click in a label cell to edit it. If no values are entered in the label row the field names F1, F2 etc will be used.

### Edit Menu

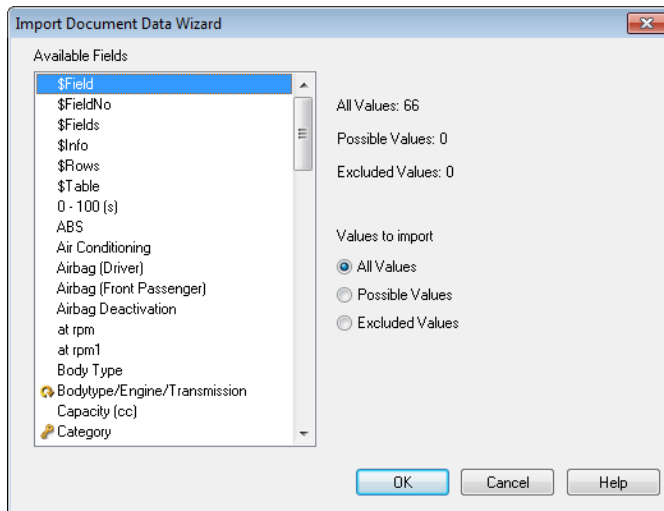
The **Edit** menu contains some basic editing commands.

- |                      |   |
|----------------------|---|
| <b>Insert Column</b> | Inserts a new blank column if one or more columns are selected. |
| <b>Delete Column</b> | Deletes the selected column(s).                                 |
| <b>Insert Row</b>    | Inserts a new blank row if one or more rows are selected.       |
| <b>Delete Row</b>    | Deletes the selected rows.                                      |

### Tools Menu

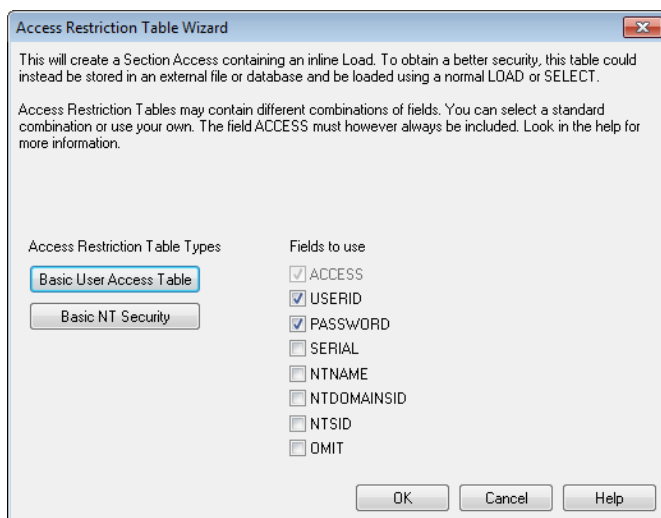
The **Tools** menu contains the command **Document Data** that opens the **Import Document Data Wizard** from which it is possible to paste field values from any existing field in the document. In the dialog it is possible to see how many field values are present in any selected field. It is also possible to choose to insert all values, selected values or excluded values into the inline wizard.

Click **OK** to close the dialog and generate a **Load inline** statement reflecting the contents of the dialog's table grid.



*The Import Document Data wizard.*

## 23.11 Access Restriction Table Wizard



*The Access Restriction Table Wizard*

The **Access Restriction Table Wizard** dialog is opened from the **Insert** menu, **Section Access, Inline**. It is used to generate a **Load Inline** statement for access control to the QlikView document. See *Security (page 417)*.

The dialog contains the following commands:

- |                                |  |
|--------------------------------|--|
| <b>Fields to Use</b>           | The list contains all the possible security fields in a QlikView <b>section access</b> . Mark the check box for those you wish to include. |
| <b>Basic User Access Table</b> | Press this button to mark ACCESS, USERID and PASSWORD while unmarking the other fields.  |
| <b>Basic NT Security</b>       | Press this button to mark ACCESS, PASSWORD, NTNAME and NTDOMAINSID while unmarking the other fields.                                       |

When closing the dialog with **OK**, the *Inline Data Wizard* (page 194) dialog 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* (page 269) statement and succeeded by a **Section Application** statement will be created.

## 23.12 File Wizard

The **File Wizard** automatically appears when opening a file or an HTML table using the **Table Files...** button in the **Data** page of the **Tool Pane** in the *Edit Script Dialog* (page 175). 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 the 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 the user from one page to another.

On the **Type** and **Options** pages there is a preview in which it is possible to see how QlikView has interpreted the file. The **Script** page makes it possible to 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.

To load a file stored on a local network, use the **Table Files** button in the **Edit Script** dialog. If the file has a table file extension, 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** dialog. 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.

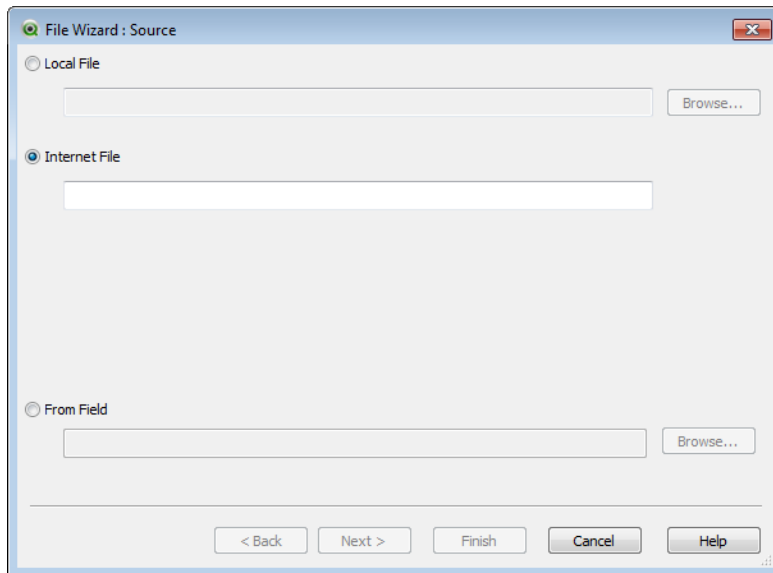
To load a file directly from the Internet, paste the URL (previously copied from your web browser) into the text box named **File Name** in the FTP browsing dialog and click **Open**.

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 script.

## File Wizard: Source

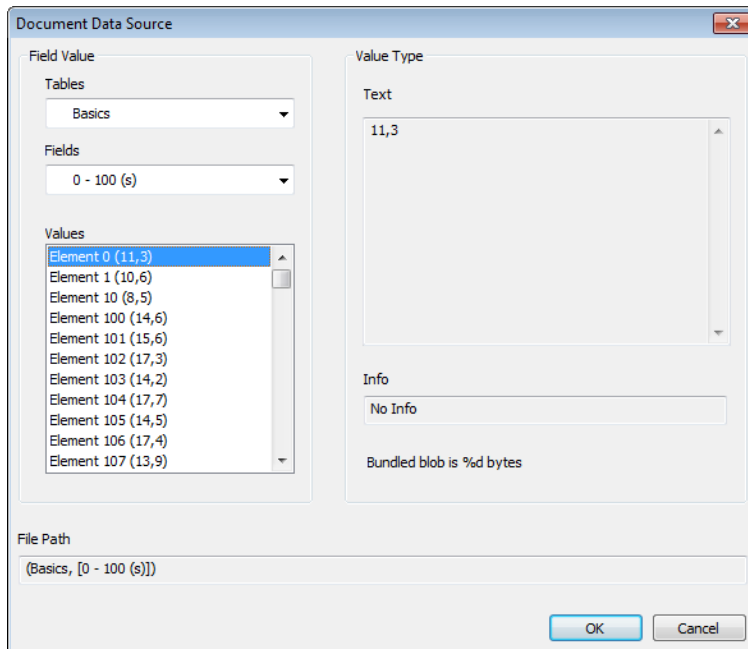


### *File Wizard, Source*

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.

- |                      |  |
|----------------------|--|
| <b>Local File</b>    | Enter the path to a local file or click <b>Browse</b> to find it through the explorer.                 |
| <b>Internet File</b> | Enter the path to the Internet file.   |
| <b>From Field</b>    | Opens the <i>Document Data Source (page 198)</i> dialog where it is possible to choose table and field |

## Document Data Source



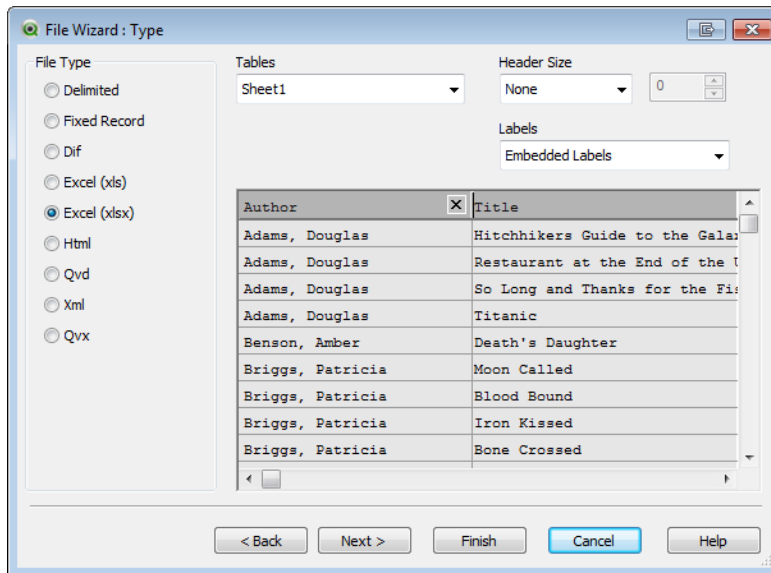
*The Document Data Source dialog*

This dialog lists the tables of the active document and makes it possible to use a previously loaded field as data source.

<b>Field Value</b>	<p><b>Tables</b> A drop-down menu showing the tables that are loaded in the active document.</p> <p><b>Fields</b> The fields of the selected table.</p> <p><b>Values</b> The values of the selected fields.</p>
<b>File Path</b>	The name of the selected table and field.
<b>Value Type</b>	<p>The type of the selected value.</p> <p><b>Text</b> If the value is text it is displayed here.</p> <p><b>Info</b> If information is connected to the field value it is displayed here.</p> <p><b>Blob</b> If a blob is connected to the value it is displayed here.</p>

## File Wizard: Type

This page contains the settings related to file type and table type. The settings vary according to file type.



*File Wizard, Type*

This page contains the settings related to file type and table type. 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.

**File Type** The type of file is set here: Delimited, Fixed record, DIF, Excel Files (xls), Excel (xlsx), HTML, QVD, XML or QVX.

**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 change 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. To rename a field, click the field name, enter the new name in this box and confirm by pressing RETURN. The **as** clauses in the **load** statement 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.

## File Wizard: Type - Delimited

*The File Wizard - Type page, delimited file*

**Delimiter** Specify what type of delimiter is to be used. To use a delimiter other than the predefined ones, choose **Custom** to find the correct delimiter. **Custom** opens the *Delimiter Selector* dialog, where it is possible to choose among the delimiters available on the computer.

<b>Header Size</b>	The header part of the file is a specified number of lines or bytes (only for delimited, fixed record and Excel files). The header may be omitted.
<b>Character Set</b>	The character set of the table file is set here. If the desired character set is not available in the drop-down menu, it is possible to choose <b>Custom</b> . This opens the <b>Codepage selector</b> dialog, where to choose among the character sets installed on the computer.
<b>Quoting</b>	Set the <i>Quotes (page 260)</i> scheme to be used. Choose between <b>Standard</b> , <b>None</b> and <b>MSQ</b> .
<b>Comment</b>	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 to enable QlikView to recognize comments.
<b>Labels</b>	If the field names (column headers) are stored in the first line of the table, this control should be set to <b>Embedded</b> . The format also allows field names to be explicitly specified, and if that is the case, this control should be set to <b>Explicit</b> . If the table contains no field names, the <b>None</b> alternative should be used.
<b>Ignore EOF</b>	Sometimes a special character is used to mark the end of the file. In case this character is found in the middle of a text file (for example enclosed by quotation marks) you can enable this option to ignore it.

## File Wizard: Type - Fixed Record

The screenshot shows a dialog box with the following controls:

- Header Size:** A dropdown menu set to 'None' and a numeric spinner set to '0'.
- Tab Size:** A numeric spinner set to '1'.
- Record Size:** A dropdown menu set to 'None' and a numeric spinner set to '0'.
- Char Set:** A dropdown menu set to 'Western European'.
- Buttons:** 'Embedded Labels...', 'Analyze Fix Positions', and 'Clear Fix Positions'.
- Checkbox:** 'Ignore EOF' (unchecked).

*The File Wizard - Type page, fixed record file*

<b>Header Size</b>	Here it is possible to omit the header part of the file: a specified number of lines or bytes (only for delimited, fixed record and Excel files).
<b>Record Size</b>	Specify the size of a record to be a specified number of lines or a specified number of bytes (only for fixed record files).
<b>Char Set</b>	The character set of the table file is set here. If the desired character set is not available in the drop-down menu, choose <b>Custom</b> . This opens the <b>Codepage selector</b> dialog, where it is possible to choose among the character sets installed on the computer.
<b>Ignore EOF</b>	Sometimes a special character is used to mark the end of the file. In case this character is found in the middle of a text file (for example enclosed by quotation marks) you can enable this option to ignore it.
<b>Tab Size</b>	The length of a tab set in spaces.
<b>Embedded Labels...</b>	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).

## File Wizard: Type - DIF

*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.

**Character Set (page 407)** The character set of the table file is set here. If the desired character set is not available in the drop-down menu, choose **Custom**. This opens the **Codepage selector** dialog, where it is possible to choose among the character sets installed on the computer.

## File Wizard: Type - Excel (XLS)

*The File Wizard - Type page, XLS file*

**Tables** In the **Tables** group it is possible to 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. Note that the Excel sheet names must not contain the characters **&**, **>** or **<**.

When defining a named area on a sheet, the selected **Scope** must be **Workbook** and the **Refers to** field must keep its original value (the one assigned by default). Otherwise, the named area might not be seen as a table.

**Header Size** Here it is possible to omit the header part of the file: 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 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.

## File Wizard: Type - Excel (XLSX)

The File Wizard - Type page, XLSX file

**Tables** In the **Tables** group it is possible to 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. Note that the Excel sheet names must not contain the characters **&**, **>** or **<**.

When defining a named area on a sheet, the selected **Scope** must be **Workbook** and the **Refers to** field must keep its original value (the one assigned by default). Otherwise, the named area might not be seen as a table.

**Header Size** Here it is possible to omit the header part of the file: 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 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.

## File Wizard: Type - HTML

The File Wizard - Type page, HTML file

**Tables** In the **Tables** group it is possible to 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 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, choose **Custom**. This opens the **Codepage selector** dialog, where it is possible to choose among the character sets installed on the computer.

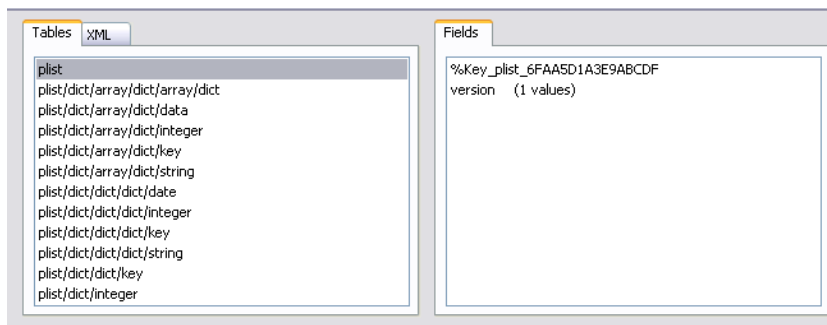
**Analyze Fix Positions** Analyzes and sets the columns widths for fix record files and for some HTML tables. After the analysis, column breaks can be added or removed in the preview pane.

**Clear Fix Positions** Clears all column breaks (for fix record files and for some HTML tables).

## File Wizard: Type - QVD

This file type has no configurable settings.

## File Wizard: Type - XML



*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.

## File Wizard: Type - QVX

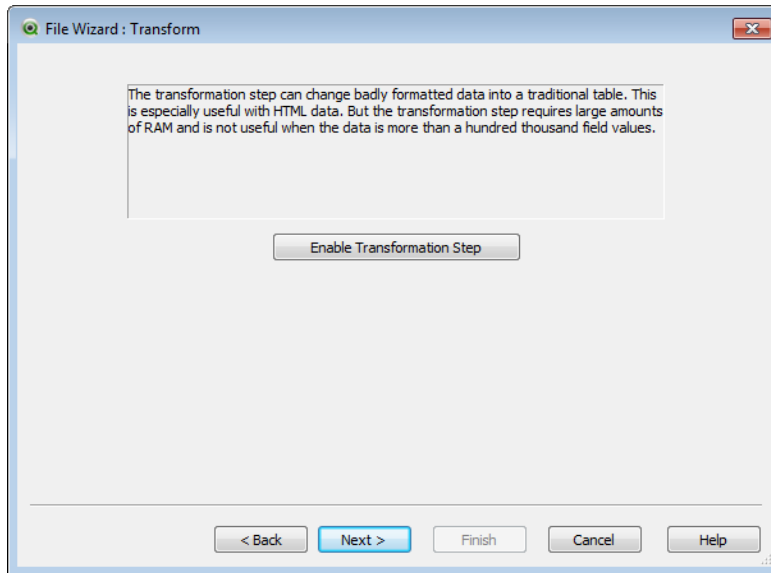
A QVX formatted file contains metadata describing a table of data and the actual data. In contrast to the QVD format, which is proprietary and optimized for minimum transformations inside QlikView, the QVX format is public and requires a few transformations when exporting data from traditional data base formats. QVX files are loaded in the script with the **load** statement.

This file type has no configurable settings.

## File Wizard: Transform

In the **Transform** dialog it is possible to filter and make advanced transformations of a table. This is particularly important for HTML table, 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.

In order to reach the **Transform** dialogs, enable the transformation step by clicking the **Enable Transformation Step** button. The step is not enabled by default in order to save memory.

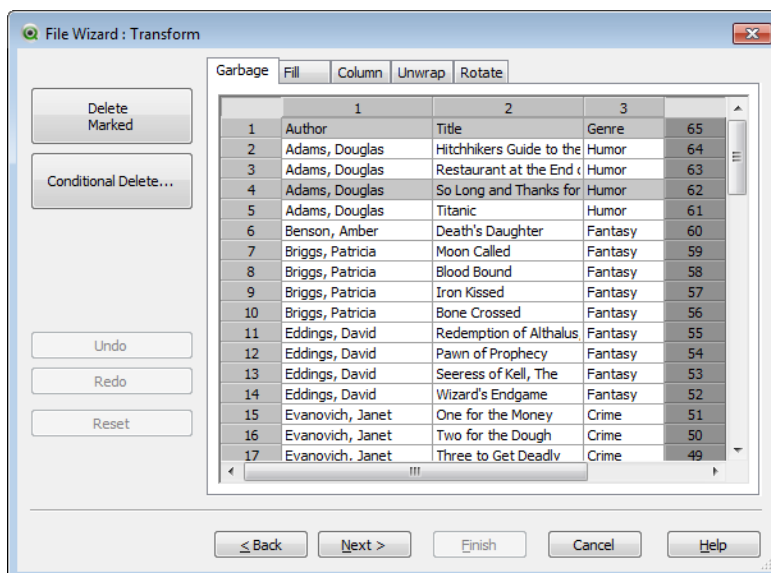


*File Wizard, Enable Transform Step*

The **Transform** dialog provides a variety of filters that can be used to make tables more suitable for Qlik-View. The dialog has six pages that are described below. The buttons to the bottom left-hand corner of **Transform Table** dialog are however used for all pages:

- Undo**                      Undoes the last added change.
- Redo**                        Redoes the last Undo.
- Reset**                        Resets the table to its original status.

## Garbage



*File Wizard: Transform, Garbage tab*

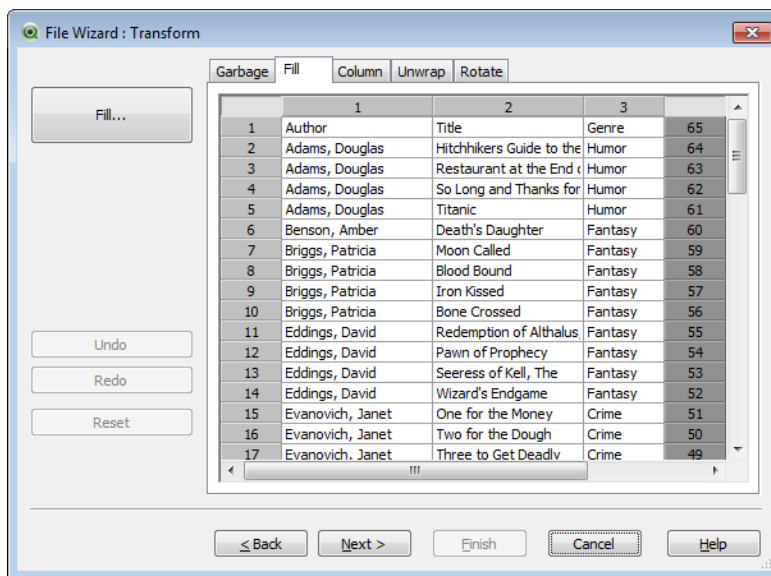
On this tab, rows and columns containing garbage can be removed from the table. In HTML files it is not uncommon to have extra rows and columns that are there just for legibility and these must of course be removed before loading the data into QlikView.

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.

**Delete Marked** Deletes the highlighted rows and columns.

**Conditional Delete...** Opens the *Specify Row Condition (page 210)* dialog where the conditional criteria for deletion of rows can be set.

## Fill

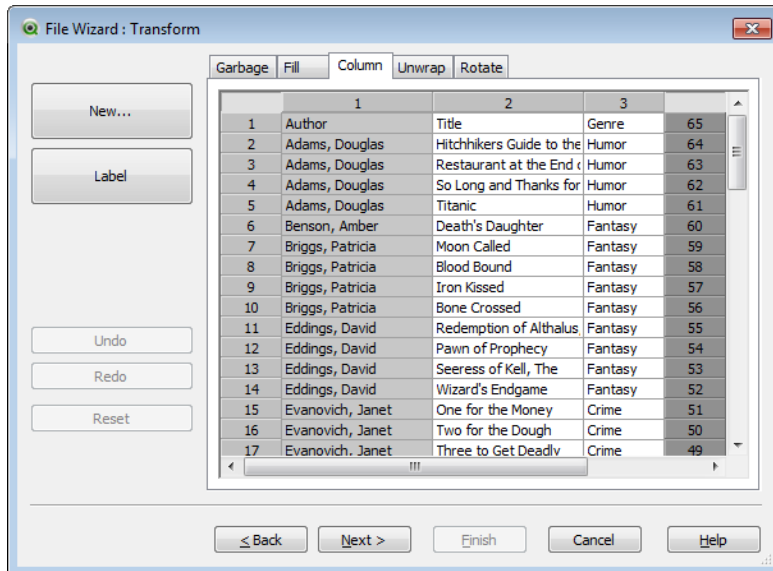


*File Wizard: Transform, Fill tab*

With this tab it is possible to replace cell values. 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...** Opens the *Fill Cells (page 211)* dialog where it is possible to specify conditions and strategy.

## Column



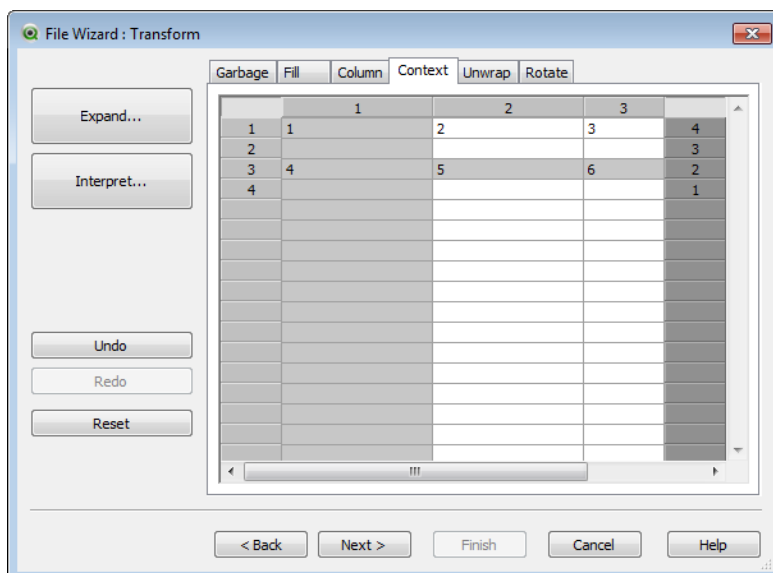
*File Wizard: Transform, Column tab*

The **Column** dialog page is used for copying the contents of a column to a new column. The copy may contain all the cells of the source column or just 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 text, whereas the other column contains its URL.

**New...** Opens the *Specify Cells for New Column (page 212)* dialog where it is possible to specify how the new column should be created.

**Label** Sets the label for the column.

## Context



*File Wizard: Transform, Context tab*

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 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>
```

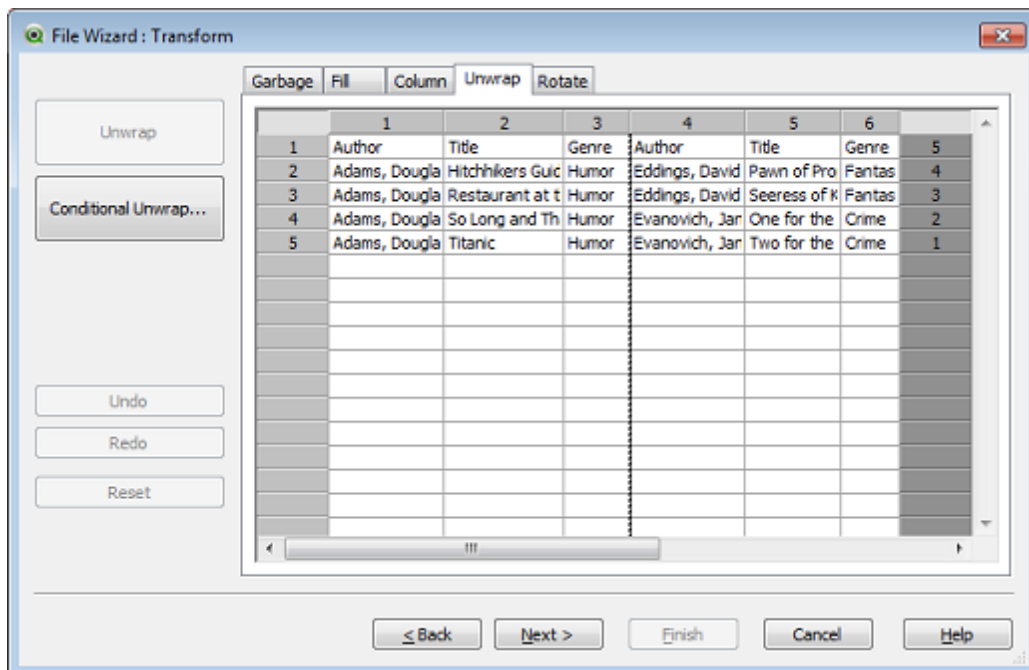
The visible text of the cell is *My link text*. The context contains a start tag and an end tag. The tags specify a hypertext link. The start tag has two attributes, *href* and *name*.

The **Context** page can be used to read the attribute of tags instead of the main string. Other kinds of operations are also possible. 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 (page 212)* dialog in which you can define an expansion of the contents of one cell into several cells. 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.

**Interpret...** Opens the *Context Interpretation (page 213)* 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.

## Unwrap



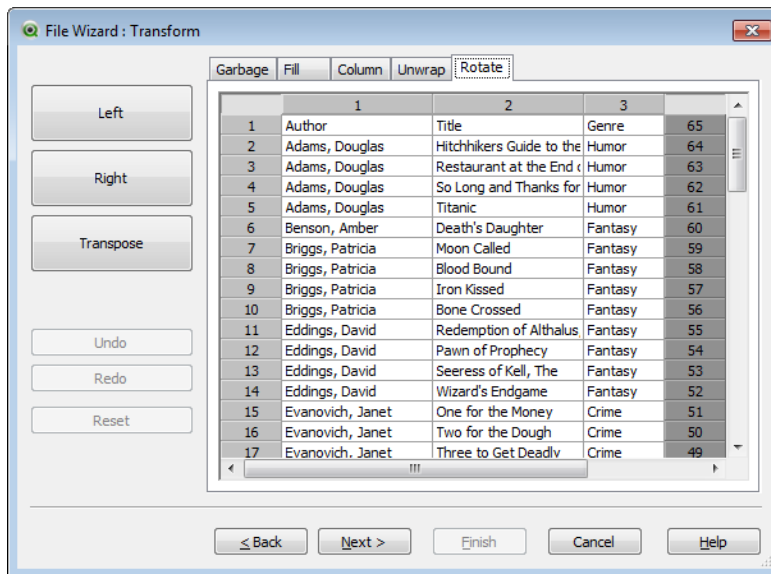
File Wizard: Transform, Unwrap tab

Here it is possible to 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* (page 210) dialog.

## Rotate



*File Wizard: Transform, Rotate tab*

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", i.e. to get fields as columns, field names on the first row etc.

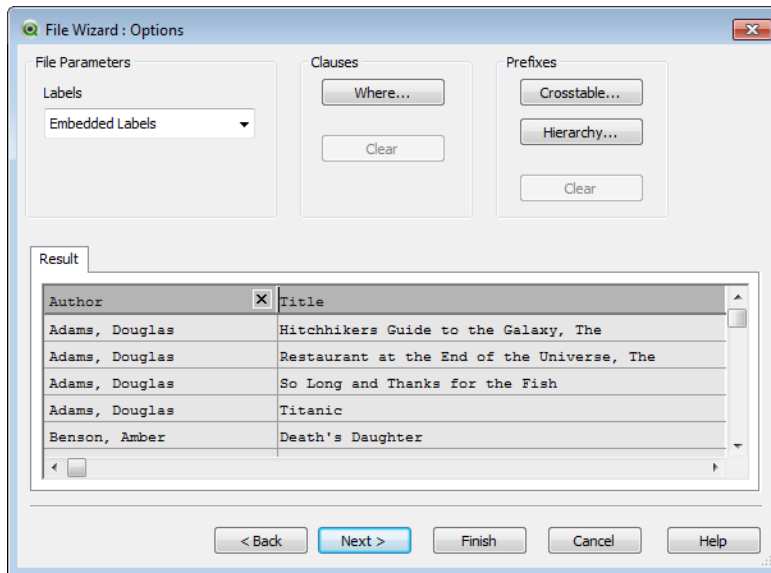
**Left** Rotates the table counterclockwise.

**Right** Rotates the table clockwise.

**Transpose** Transposes the table, i.e. mirrors the table along a diagonal axis, so that the top right cell becomes the bottom left cell and vice versa. The top left and bottom right cells, however, stay in place.



## File Wizard: Options



*File Wizard, Options*

## 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 (page 214)* dialog, 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 *Crosstable (page 229)* dialog 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 (page 215)* dialog, where it is possible to set the parameters for a hierarchy table (parameters as they appear in the script in parentheses).

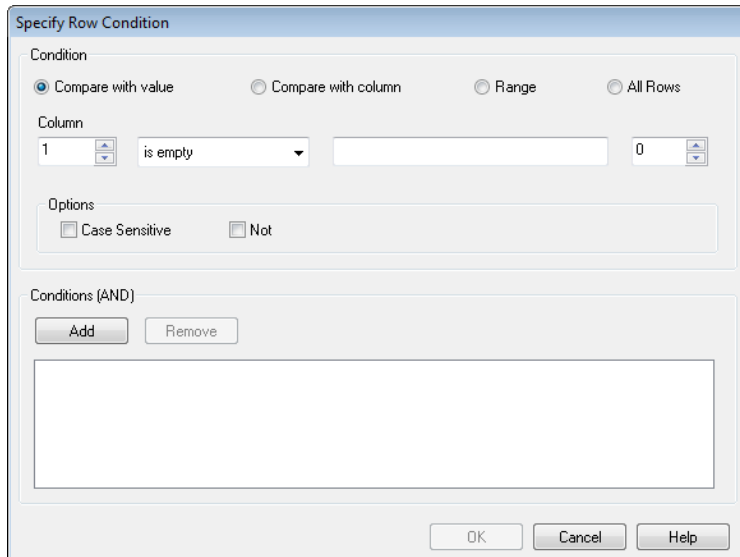
**Clear** Revokes the transformation of a cross table or a hierarchy table.

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;
```

## Specify Row Condition



*The Specify Row Condition dialog*

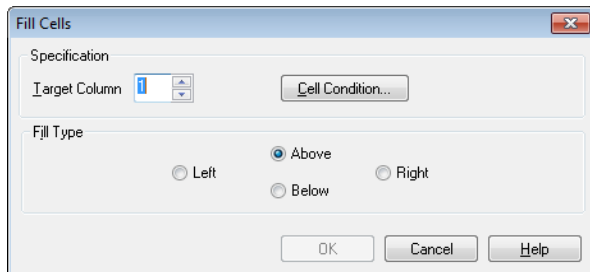
This dialog can be opened via the **File Wizard : Transform** dialog's **Garbage** page (the **Conditional Delete** button), the **Column** page (the **New...** button) or the **Unwrap** page (the **Conditional Unwrap** button). Here it is possible to define a logical condition for selection of one or 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.

In the **Condition** group it is possible to 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 press the **Add** button to include it in the transformation.

<b>Compare with value</b>	Matches the cell with a cell condition that can evaluate to true or false.
<b>Compare with column</b>	Matches the cell with the corresponding cell in another column.
<b>Range</b>	Selects x rows, then skips y rows. Starting and ending positions for this may be specified with indexes.
<b>All Rows</b>	Use this condition if all rows should be selected.
<b>Column</b>	Sets the column number on which the condition should be applied.
<b>From</b>	Only visible in <b>Range</b> mode. Sets the row number from which rows should be deleted or a second condition should be applied.
<b>To</b>	Only visible in <b>Range</b> mode. Sets the row number to which rows should be deleted or a second condition should be applied.
<b>Select</b>	Only visible in <b>Range</b> mode. Here it is possible to set whether rows should be selected or skipped cyclically, e.g. cyclically select two rows, then skip one. If all rows should be used, <b>Select</b> should be set to 1 and <b>Skip</b> should be set to 0.

- Options** In this subgroup it is possible to specify modifiers for the selection.
- Case Sensitive**  
Enable this option if comparisons should be case sensitive.
  - Not**  
Enable this option if the selection criteria should be inverted.
- Conditions (AND)** In this group the conditions specified and included in the transformation are seen. A logical AND is applied between conditions.
- Add**  
Adds the current condition to the list.
  - Remove**  
Removes a selected condition from the list.

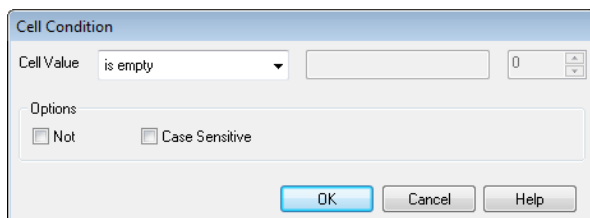
## Fill Cells



*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 (page 211)* 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



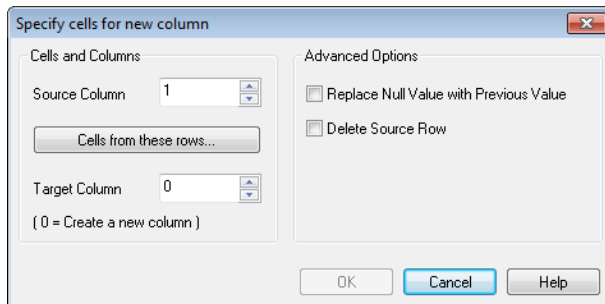
*The Cell Condition dialog*

This dialog opens from the *Fill Cells (page 211)* 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 the value comparison should be case sensitive.

## Specify Cells for New Column

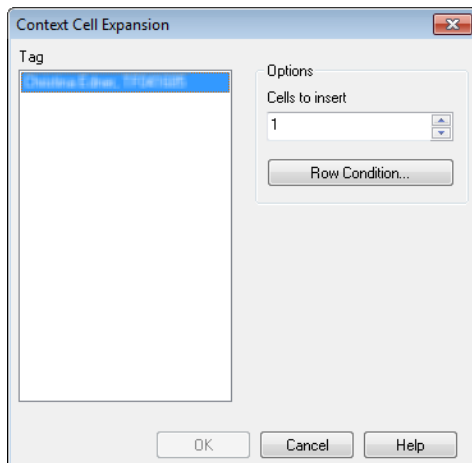


*The Specify Cells for New Column dialog*

This dialog is used for specifying the conditions for automatically generating a new column from an existing one. It is accessed from the **Column** page in the **File Wizard : Transform** dialog.

- |                          |   |
|--------------------------|---|
| <b>Cells and Columns</b> | <p><b>Source Column</b><br/>Sets the column number from which cell values should be copied.</p> <p><b>Cells from these rows</b><br/>Opens the <i>Specify Row Condition (page 210)</i> dialog where it is possible to set the row condition.</p> <p><b>Target Column</b><br/>Specifies the number of the target columns.</p> |
| <b>Advanced Options</b>  | <p><b>Replace Null Value with Previous Value</b><br/>Fills empty cells with the value in the cell above.</p> <p><b>Delete Source Row</b><br/>Deletes the source row after copying its content.</p>  |

## Context Cell Expansion



*The Context Cell Expansion dialog*

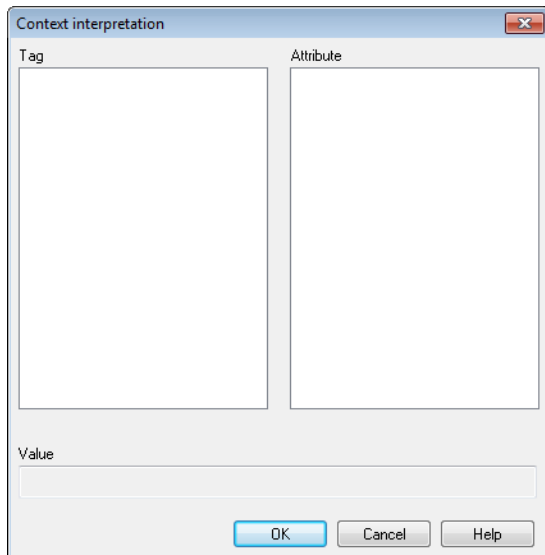
With this dialog it is possible to expand the contents of one cell into several cells. In some situations several rows in a column are stored within one cell of the HTML table, i.e. between the <TD> and the </TD> tags.

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 <BR> symbol.

**Cells to Insert**                      Number of cells to insert. Normally this should be set to 1, but if extra rows are needed, this control can be set to a higher number.

**Row Condition**                      Condition for what rows to expand.

## Context Interpretation



### *The Context Interpretation dialog*

With this dialog it is possible to exchange the value of a cell with a 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.

## Where Clause

The Where Clause dialog

- Simple** Choose what **Field(s)** should be part of the where clause and what **Operator/Function** should be used. It is also possible to enter a **Constant** based on existing field values. The second and third rows become active when selecting **AND** or **OR** in the drop-down list to the left of each row.
- Advanced** Mark **Advanced** to type the clause from scratch.
- Empty Template** This option will insert **Where (1=1)** into the script and it is possible to edit it further there.

## Crosstable

A cross table is a common type of table featuring a matrix of values between two or more orthogonal lists of header data, of which one is used as column headers. A typical example could be to have one column per month. To transform a cross table into a straight table, use a **crosstable** prefix. The result is that the column headers (e.g. month names) will be stored in one field - the attribute field - and the column data (month numbers) will be stored in a second field: the data field.

The syntax is:

```
crosstable (attribute field name, data field name [ , n ] ) ( load-statement | selectstatement )
```

where:

*attribute field name* is the field to contain the attribute values.

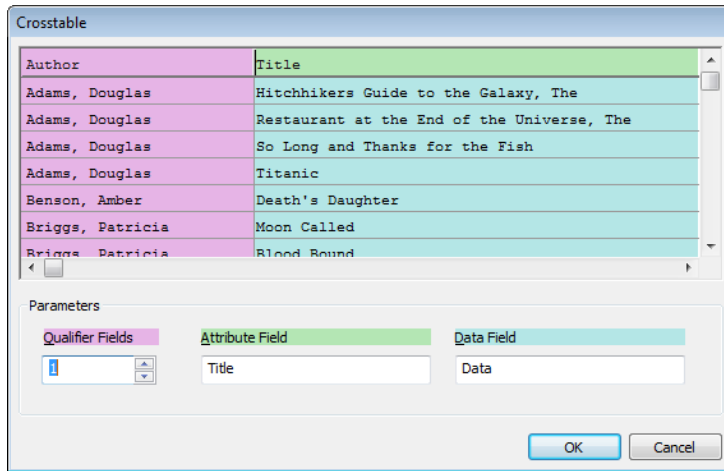
*data field name* is the field to contain the data values.

*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;
```

## Crosstable Wizard

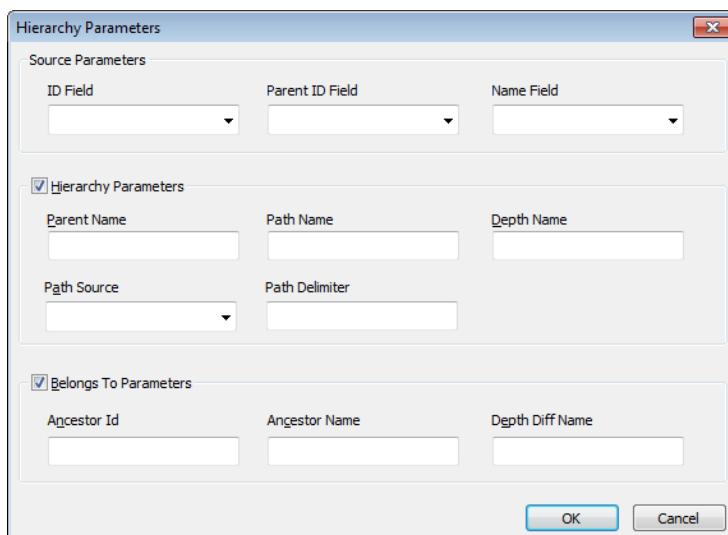


### *The Crosstable wizard*

The crosstable wizard is dialog driven method of creating the crosstable statement. This dialog is opened by clicking the **Crosstable** button in the **Options** page of the **File Wizard**. The crosstable wizard holds the following options:

- Qualifier Fields**            The number of qualifier fields that precede the fields to be transformed.
- Attribute Field**            The name of the new field that will contain all the fields (attribute values) to be transformed.
- Data Field**                 The name of the new field that will contain the data of the attribute values.

## Hierarchy Parameters



### *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.

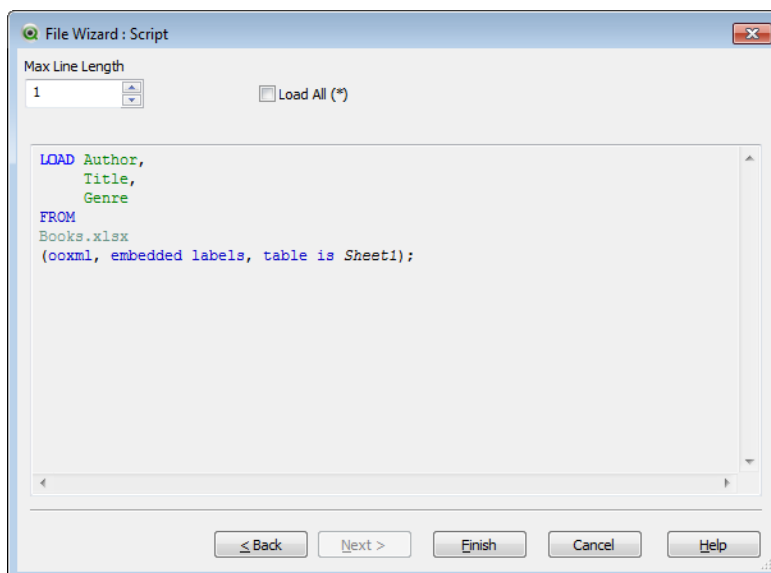
**Ancestor Name** (*AncestorName*)

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.

## File Wizard: Script



*File Wizard, Script*

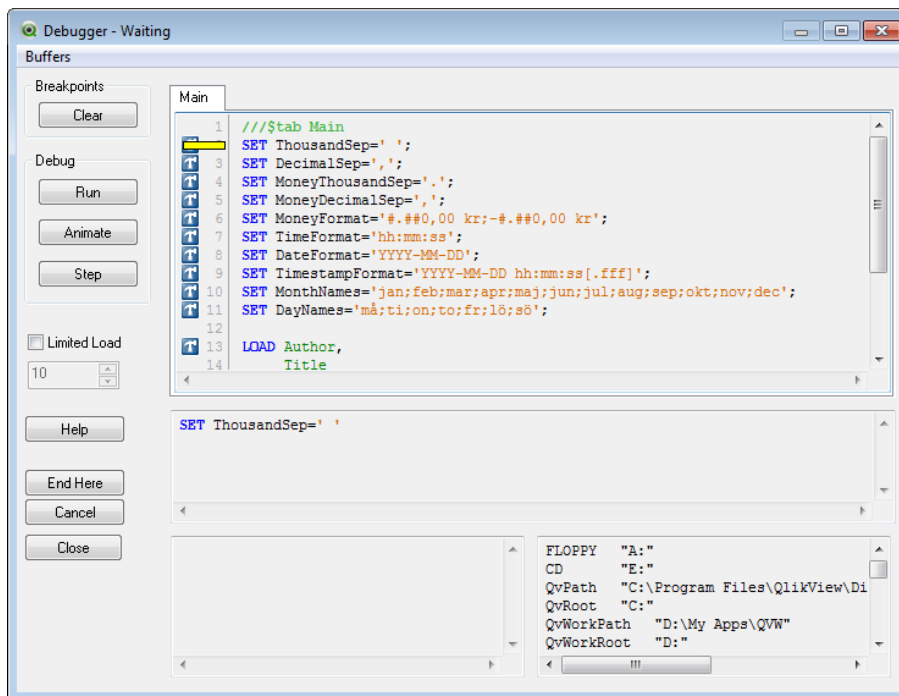


The **Script** page makes it possible to set how the script should be presented in the **Edit Script** dialog. Here it is also possible to set a **Max Line Length** to make the script easier to interpret.

**Load All (\*)** loads all fields in the table.

The **Finish** button generates the **load** statement and closes the wizard.

## Debugger



### The Script Debugger

Running the script in the debugger can make it a lot easier to identify script errors. The debugger makes it possible to monitor every script statement and examine the variable values while the script is executed.

The script is displayed in the window in the upper half of the debugger. The progress of the script execution is marked with a yellow bar. **Break points** can be inserted by clicking on the row numbers and removed by clicking a second time. Remove all breakpoints by clicking the **Clear** button. When a breakpoint is encountered, script execution is halted until instructed to proceed.

The statement being currently executed is displayed in the window in the middle.

Status codes and any script errors are displayed in the window to the bottom, left. This is essentially the same information that is displayed in the **Script Execution Progress** window when the script is run outside the debugger.

The right bottom window displays all *Variable (page 173)* and their respective values. Variables that have changed are colored red.

#### Breakpoints

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.

<b>Debug</b>	<p>This group contains settings that determine how the script is run in the Debugger.</p> <p><b>Run</b> This is the normal script execution. The script will progress to the end or until a breakpoint is encountered.</p> <p><b>Animate</b> The script runs as above, but with a short pause after each statement. This way it is possible to monitor the script execution more closely.</p> <p><b>Step</b> This executes one script statement at a time.</p>
<b>Limited Load</b>	<p>Enter a number in the spinner box below. The number entered is the maximum number of accepted records at each <b>load</b> and <b>select</b> statement. This is a very useful way to limit the execution time when the script is tested with live data.</p>
<b>Help</b>	<p>Opens the QlikView help.</p>
<b>End Here</b>	<p>Closes the <b>Debugger</b>, but keeps the data loaded so far.</p>
<b>Cancel</b>	<p>Aborts the script execution and discards all data loaded.</p>
<b>Buffers</b>	<p>Opens tabs displaying external script files referenced via <b>\$(include)</b>.</p> <p><b>Open</b> Displays a dialog from which a script file can be opened. The contents of the file will be displayed under a separate tab and can be used for debugging.</p> <p><b>Close</b> Closes the current debugger script tab. The main tab cannot be closed.</p>

## 24 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 *Backus-Naur Formalism* (page 927).

### 24.1 Script Statements and Keywords

The QlikView script consists of a number of statements. A statement can be either a regular script statement or a *Script Control Statements* (page 221). Certain statements can be preceded by prefixes.

Regular statements are typically used for manipulating 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, an alphabetical listing of all script statements, control statements and prefixes, are found.

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.

Available script statements are:

<b>Add</b>	<i>page 222</i>
<b>Alias</b>	<i>page 223</i>
<b>Binary</b>	<i>page 223</i>
<b>Buffer</b>	<i>page 223</i>
<b>Bundle</b>	<i>page 224</i>
<b>Call</b>	<i>page 225</i>
<b>Comment Field</b>	<i>page 226</i>
<b>Comment Table</b>	<i>page 226</i>
<b>Concatenate</b>	<i>page 227</i>
<b>Connect</b>	<i>page 228</i>
<b>Crosstable</b>	<i>page 229</i>
<b>Direct Query</b>	
<b>Directory</b>	<i>page 236</i>
<b>Disconnect</b>	<i>page 236</i>
<b>Do..loop</b>	<i>page 236</i>
<b>Drop Field</b>	<i>page 237</i>

<b>Drop Table</b>	<i>page 237</i>
<b>Execute</b>	<i>page 238</i>
<b>Exit Script</b>	<i>page 238</i>
<b>First</b>	<i>page 238</i>
<b>For..next</b>	<i>page 239</i>
<b>For each..next</b>	<i>page 240</i>
<b>Force</b>	<i>page 241</i>
<b>Generic</b>	<i>page 242</i>
<b>Hierarchy</b>	<i>page 242</i>
<b>HierarchyBelongsTo</b>	<i>page 243</i>
<b>If..then..elseif..else..end if</b>	<i>page 244</i>
<b>Image_size</b>	<i>page 245</i>
<b>Info</b>	<i>page 245</i>
<b>Inner</b>	<i>page 245</i>
<b>Inputfield</b>	<i>page 246</i>
<b>IntervalMatch</b>	<i>page 247</i>
<b>IntervalMatch (Extended Syntax)</b>	<i>page 249</i>
<b>Join</b>	<i>page 250</i>
<b>Keep</b>	<i>page 250</i>
<b>Left</b>	<i>page 251</i>
<b>Let</b>	<i>page 252</i>
<b>Load</b>	<i>page 253</i>
<b>Loosen Table</b>	<i>page 260</i>
<b>Mapping</b>	<i>page 261</i>
<b>Map ... using</b>	<i>page 261</i>
<b>NoConcatenate</b>	<i>page 262</i>
<b>NullAsNull</b>	<i>page 263</i>
<b>NullAsValue</b>	<i>page 262</i>
<b>Outer</b>	<i>page 263</i>
<b>Qualify</b>	<i>page 264</i>
<b>Rem</b>	<i>page 265</i>
<b>Rename Field</b>	<i>page 265</i>

<b>Rename Table</b>	<i>page 266</i>
<b>Replace</b>	<i>page 268</i>
<b>Right</b>	<i>page 267</i>
<b>Sample</b>	<i>page 269</i>
<b>Section</b>	<i>page 269</i>
<b>Select (SQL)</b>	<i>page 270</i>
<b>Semantic</b>	<i>page 272</i>
<b>Set</b>	<i>page 272</i>
<b>Sleep</b>	<i>page 272</i>
<b>SQL</b>	<i>page 272</i>
<b>SQLColumns</b>	<i>page 273</i>
<b>SQLTables</b>	<i>page 273</i>
<b>SQLTypes</b>	<i>page 274</i>
<b>Star</b>	<i>page 275</i>
<b>Store</b>	<i>page 276</i>
<b>Sub..end sub</b>	<i>page 276</i>
<b>Switch..case..default..end switch</b>	<i>page 277</i>
<b>Tag Field</b>	<i>page 278</i>
<b>Trace</b>	<i>page 279</i>
<b>Unless</b>	<i>page 279</i>
<b>Unmap</b>	<i>page 280</i>
<b>Unqualify</b>	<i>page 280</i>
<b>Untag Field</b>	<i>page 280</i>
<b>When</b>	<i>page 281</i>

## Script Control Statements

The QlikView script consists of a number of statements. A statement can be either a regular *Script Statements and Keywords* (page 219) or a script control statement.

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 semicolon or end-of-line.

Prefixes are never applied to control statements, with the exceptions of the prefixes **when** and **unless** which may be used with a few specific control statements.

All script keywords can be typed with any combination of lower case and upper case characters.

Available script control statements are:

<b>If..then..elseif..else..end if</b>	<i>page 244</i>
---------------------------------------	-----------------

<b>For..next</b>	<i>page 239</i>
<b>For each..next</b>	<i>page 240</i>
<b>Do..loop</b>	<i>page 236</i>
<b>Switch..case..default..end switch</b>	<i>page 277</i>
<b>Exit Script</b>	<i>page 238</i>
<b>Sub..end sub</b>	<i>page 276</i>
<b>Call</b>	<i>page 225</i>

## Add

The **add** prefix can be added to any *Load* (page 253), *Select (SQL)* (page 270) or *Map ... using* (page 261) statement in the script. It is only relevant during *partial reloads*. During a partial reload the QlikView table, for which a table name is generated by the **add load/add select** statement (provided 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 **map...using** statement causes mapping to take place also during partial script execution.

The syntax is:

```
add [only] (loadstatement |selectstatement |mapstatement)
```

where:

**only** is an optional qualifier denoting 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. See *concatenated*.

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 *exists(field [ , expression ])* (page 338)).

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 seeing if *Name* exists in the previously loaded table data.

```
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 seeing if *Name* exists in the previously loaded table data.

Back to **Script Statements and Keywords**.

## Alias

The **alias** statement is used for setting 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* where *fieldname* represent the name of the field in your source data, and *aliasname* represent an alias name that you want to use instead.

### 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 (SQL)* (page 270) and *Load* (page 253) statements. A new alias can be defined for a field name by a new **alias** statement at any subsequent position in the script.

Back to *Script Statements and Keywords* (page 219).

## Binary

The **binary** statement is used for loading the data from another QlikView document, including section access data. It does not load the layout information or variables. 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.

Back to *Script Statements and Keywords* (page 219).

## Buffer

QVD files can be created and maintained automatically via the **buffer** prefix. This prefix can be used on most *Load* (page 253) and *Select (SQL)* (page 270) statements in script. It indicates that *QVD Files* (page 409) are used to cache/buffer the result of the statement.

Numerous limitations exist, most notable is that there must be either a file **load** or a **select** statement at the core 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 85)*. 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 anywhere throughout a complete script execution in the document that created it or when the document that created it no longer exists. This house-keeping can be turned off by checking the options **Keep Unreferenced QVD Buffers** in *Document Properties: General (page 424)* (first requirement) and **Keep Orphaned QVD Buffers** in *User Preferences (page 69)* (second requirement) property pages respectively, though the procedure cannot be generally recommended.

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 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. Note that the **incremental** option can only be used with **load** statements and text files and that **incremental load** cannot be used where old data is changed or deleted!

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 QVD buffer after which it will no longer be considered valid. Before that time the QVD buffer 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 SQL SELECT * FROM MyTable;

buffer (stale after 7 days) SQL SELECT * FROM MyTable;

buffer (incremental) LOAD * FROM MyLog.log;
```

Back to *Script Statements and Keywords (page 219)*.

## Bundle

The **Bundle** prefix is used to include external files, such as image or sound files, or objects connected to a field value, to be stored in the qvw file.

If a piece of external information, such as an image or a sound file is to be connected to a field value, this is done in a table that is loaded with the **Info** prefix. In order to maintain portability, it is possible to include the external files into the .qvw file itself. To this end use the **Bundle** prefix. The bundled info files are compressed in the process, but will nevertheless take up additional space both in the file and in RAM. Therefore consider both the size and the number of bundled files before going for this solution.

The info may be referenced from the layout as normal info, via the chart info function or 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.

```
Bundle [Info] ( loadstatement | selectstatement )
```

### Examples:

```
Bundle Info Load * Fom flagoecd.csv;
Bundle SQL Select * From infotable;
```

## Call

The **call** control statement calls a subroutine which must be defined by a previous **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 or an arbitrary expression.

The subroutine called by a **call** statement must be defined by a **sub** encountered earlier during script execution.

Parameters are copied into the subroutine and, if the parameter in the **call** statement is a variable and not an expression, copied back out again upon exiting the subroutine.

Since the **call** statement is a control statement and as such is ended with either a semicolon or end-of-line, it must not cross a line boundary.

### 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:')
```

Back to *Script Statements and Keywords (page 219)*.

## Comment Field

Provides a way of displaying the field comments (metadata) from databases and spreadsheets. Field names not present in the document will be ignored. If multiple occurrences of a field name are found, the last value is used.

The keyword can be used to read comments from a data source using the following syntax:

```
comment *fieldlist using mapname
```

*\*fieldlist* is a comma separated list of the fields to be commented. Using \* as field list indicates all fields. The wildcard characters \* and ? are allowed in field names. Quoting of field names may be necessary when wildcards are used.

*mapname* is the name of a mapping table previously read in a mapping load or mapping select statement (see *Mapping (page 261)*).

The map table used should have two columns, the first containing field names and the second the comments. To set individual comments the following syntax is used:

```
comment fieldname with comment
```

*fieldname* is the name of the field that should be commented.

*comment* is the comment that should be added to the field.

### Example 1:

```
commentmap:
mapping load * inline [
a,b
Alpha,This field contains text values
Num,This field contains numeric values
];
comment fields using commentmap;
```

### Example 2:

```
comment field Alpha with AFieldContainingCharacters;
comment field Num with A field containing numbers';
comment Gamma with 'Mickey Mouse field';
```

Back to *Script Statements and Keywords (page 219)*.

## Comment Table

Provides a way of displaying the table comments (metadata) from databases or spreadsheets. Table names not present in the document are ignored. If multiple occurrences of a table name is found, the last value is used.

The keyword can be used to read comments from a data source using the syntax:

```
comment tables tablelist using mapname
```

---

where

*tablelist* := (*table*{*table*})

*mapname* is the name of a mapping table previously read in a mapping load or mapping select statement (see *Mapping (page 261)*).

To set individual comments, the following syntax is used:

```
comment tables tablename with comment
```

where

*tablename* is the name of the table that should be commented.

*comment* is the comment that should be added to the table.

### Example 1:

```
Commentmap:
mapping Load * inline [
a,b
Main,This is the fact table
Currencies, Currency helper table
];
comment tables using commentmap;
```

### Example 2:

```
comment table Main with 'Main fact table';
```

Back to *Script Statements and Keywords (page 219)*.

## Concatenate

If two tables that are to be concatenated have different sets of *Fields (page 171)*, *concatenation* of two tables can still be forced with the **Concatenate** prefix. This statement forces concatenation with an existing named table or the latest previously created *logical table*. A concatenation is in principle the same as the SQL UNION statement, but with two differences: first that **Concatenate** prefix can be used no matter if the tables have identical field names or not; and secondly that no removal of identical records are made.

The syntax is:

```
concatenate [ (tablename ) ] ( loadstatement | selectstatement )
```

### Examples:

```
Concatenate LOAD* FROM file2.csv;
Concatenate SQL SELECT * FROM table3;
tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
. . . .
Concatenate (tab1) load * from file3.csv;
```

Back to *Script Statements and Keywords (page 219)*.

## Connect

A general database can be accessed through the OLE DB/ODBC interface. For ODBC, the data source first needs to be specified using the ODBC administrator (see Windows Control Panel). Which database to use is in QlikView defined through the **CONNECT** statement. The syntax is:

```
ODBC CONNECT TO connect-string [ ( access_info ) ]
```

```
OLEDB CONNECT TO connect-string [ ( access_info ) ]
```

```
CUSTOM CONNECT TO connect-string
```

where:

```
connect-string ::= datasourcename { ; conn-spec-item }
```

The connection string is the data source name and an optional list of one or more connection specification items. If the data source name contains blanks, or if any connection specification items are listed, the connection string must be enclosed by *Quotation Marks in Scripting* (page 292).

*datasourcename* must be a defined ODBC data source or a string that defines an OLE DB provider.

```
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, also other items than the above are possible. For OLE DB, some of the connection specific items are mandatory and not optional.

```
access_info ::= access_item { , access_item }
```

```
access_item ::= ( userid is userid | xuserid isscrambledusername ) | ( password is password | xpassword isscrambledpassword ) | codepage is codepageID | mode iswrite
```

```
codepageID ::= ansi|oem|unicode| codepagenumber
```

If the connect statement is generated by the provided wizard any user ID and password will be generated with the scrambled **xuserid is** / **xpassword is** syntax, provided that **Scramble Connect User Credentials** is checked in *User Preferences* (page 69). The non-scrambled **userid is** / **password is** syntax must be used for providing user ID and password, if typing the connect statement. Full scrambling is currently only available for **ODBC connect** statements as some parts of the **OLEDB connect** string cannot be scrambled. Note that the scrambling is not a proper security measure; it is merely an obfuscation of the credentials.

The **codepage** specifier can be used if problems are encountered 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.

If the **ODBC** is placed before **CONNECT**, the ODBC interface will be used; else, OLE DB will be used.

### Example:

```
ODBC CONNECT TO 'Nwind;
DBQ=C:\Program Files\Access\Samples\Northwind.mdb' (UserID is sa, Password is admin);
```

The data source defined through this statement is used by subsequent *Select (SQL)* (page 270) statements, until a new **CONNECT** statement is made.

How the connect statement connects to the database, using a 32-bit or 64-bit provider, is determined by what version of QlikView and Windows that are running:

Windows	QlikView	Connect statement	Result
64-bit	64-bit	connect	uses a 64-bit ODBC
64-bit	64-bit	connect32	uses a 32-bit ODBC

64-bit	64-bit	connect64	uses a 64-bit ODBC
64-bit	32-bit	connect	uses a 32-bit ODBC
64-bit	32-bit	connect32	uses a 32-bit ODBC
64-bit	32-bit	connect64	uses a 64-bit ODBC
32-bit	32-bit	connect	uses a 32-bit ODBC
32-bit	32-bit	connect32	uses a 32-bit ODBC
32-bit	32-bit	connect64	causes script error

### **Connect32**

This statement is used the same way as the **Connect** statement, but forces a 64-bit system to use a 32-bit ODBC/OLEDB provider. Not applicable for custom connect.

### **Connect64**

This statement is used the same way as the **Connect** statement, but forces use of a 64-bit provider. Not applicable for custom connect.

Back to *Script Statements and Keywords (page 219)*.

## **Crosstable**

A cross table is a common type of table featuring a matrix of values between two or more orthogonal lists of header data, of which one is used as column headers. A typical example could be to have one column per month. To transform a cross table into a straight table, use a **crosstable** prefix. The result is that the column headers (e.g. month names) will be stored in one field - the attribute field - and the column data (month numbers) will be stored in a second field: the data field.

The syntax is:

```
crosstable ( attribute field name, data field name [ , n ] ) ( load-statement | selectstatement )
```

where:

*attribute field name* is the field to contain the attribute values.

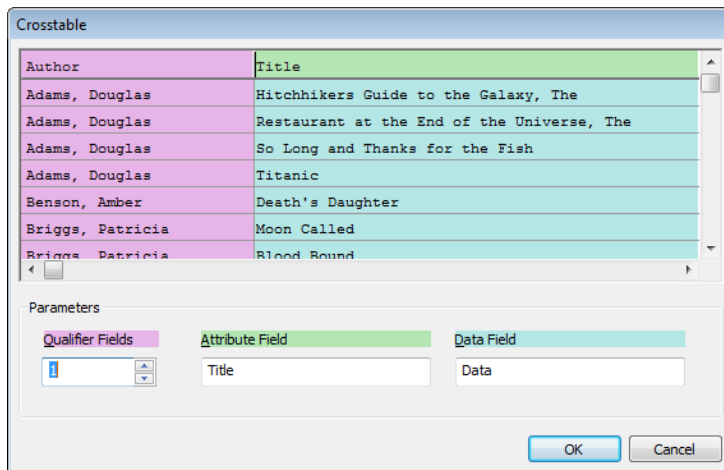
*data field name* is the field to contain the data values.

*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;
```

## Crosstable Wizard



### The Crosstable wizard

The crosstable wizard is dialog driven method of creating the crosstable statement. This dialog is opened by clicking the **Crosstable** button in the **Options** page of the **File Wizard**. The crosstable wizard holds the following options:

- Qualifier Fields**      The number of qualifier fields that precede the fields to be transformed.
- Attribute Field**      The name of the new field that will contain all the fields (attribute values) to be transformed.
- Data Field**            The name of the new field that will contain the data of the attribute values.

## Direct Query

The **DIRECT QUERY** statement allows you to link a table through an *ODBC* or *OLE DB* connection using the *Direct Discovery* (page 161) function.

```
DIRECT QUERY DIMENSION fieldlist [MEASURE fieldlist] [DETAIL fieldlist] | [DETACH
fieldlist]FROM tablename
[WHERE where_clause]
```

The **DIMENSION**, **MEASURE**, and **DETAIL** keywords can be used in any order.

The **DIMENSION** and **FROM** keyword clauses are required on all **DIRECT QUERY** statements. The **FROM** keyword must appear before the **DIMENSION** keyword.

---

Note: The **STALE** keyword no longer exists in the Direct Query syntax. The caching limit to the Direct Discovery query results is now set by the *DirectCacheSeconds* (page 234) script variable.

---

The fields specified directly after the **DIMENSION** keyword are loaded in memory and can be used to create associations between in-memory and Direct Discovery data.

---

Note:

The **DIRECT QUERY** statement cannot contain **DISTINCT** or **GROUP BY** clauses.

---

Using the **MEASURE** keyword, you can define which fields, besides the linked fields listed after the **DIMENSION** keyword, are available in QlikView. **MEASURE** fields are recognized on a "meta level." **MEASURE** field data resides only in the database.

---

The *tablename* is the name of the table or view in the database from which data will be loaded. Typically, it will be a view that contains a JOIN performed on the database.

Typically, fields with discrete values that will be used as dimensions should be loaded with the DIMENSION keyword, whereas numbers that will be used in aggregations only should be selected with the MEASURE keyword.

DETAIL fields provide information or details, like "Comment" fields, that a user may want to display in a drill-to-details table box. DETAIL fields cannot be used in chart expressions.

DETACH fields are like DIMENSION fields except that association queries are not formulated when a DETACH is selected in a List Box or Chart. For example:

```
Direct Query

    DIMENSION A
    MEASURE X, Y, Z
    DETACH B, C

From MySalesTable;
```

In this example, List Boxes for the B and C fields are grayed out; no green or white selections will show in those List Boxes. Only List Boxes or Charts with DIMENSION A are updated when selections are made.

Examples:

In this example, a database table called TableName, containing fields Dim1, Dim2, Num1, Num2 and Num3, is used. Dim1 and Dim2 will be loaded into the QlikView dataset.

```
DIRECT QUERY DIMENSTION Dim1, Dim2 MEASURE Num1, Num2, Num3 FROM TableName
```

Dim1 and Dim2 will be available for use as dimensions. Num1, Num2 and Num3 will be available to QlikView for aggregations. Dim1 and Dim2 are also available for aggregations. The type of aggregations for which Dim1 and Dim2 can be used depends on their data types. For example, in many cases DIMENSION fields contain string data such as names or account numbers. Those fields cannot be summed, but they can be counted: `count (Dim1)`.

Note:

DIRECT QUERY statements are written directly in the Edit Script dialog box. They are not generated with the Create Select Statement dialog box the way SELECT statements are. To simplify construction of DIRECT QUERY statements, you can generate a SELECT statement with the Create Select Statement dialog box and then edit the generated script to change it into a DIRECT QUERY statement.

For example, the SELECT statement:

```
SQL SELECT
    SalesOrderID,
    RevisionNumber,
    OrderDate,
    SubTotal
    TaxAmt
FROM AdventureWorks.Sales.SalesOrderHeader;
```

could be changed to the following DIRECT QUERY statement:

```
DIRECT QUERY
    DIMENSION
        SalesOrderID,
        RevisionNumber,

    MEASURE
```

```
SubTotal
TaxAmt
DETACH
OrderDate,
FROM AdventureWorks.Sales.SalesOrderHeader;
```

---

## Field lists

A field list is a comma-separated list of field specifications, *fieldname {, fieldname}*. A field specification can be a field name, in which case the same name is used for the database column name and the QlikView field name. Or a field specification can be a "field alias," in which case a database expression or column name is given a QlikView field name.

Field names can be either simple names or quoted names. A simple name begins with an alphabetic Unicode character and is followed by any combination of alphabetic or numeric characters or underscores. Quoted names begin with a double quotation mark and contain any sequence of characters. If a quoted name contains double quotation marks, those quotation marks are represented using two adjacent double quotation marks.

QlikView field names are case-sensitive. Database field names may or may not be case-sensitive, depending on the database. Direct Discovery Query preserves the case of all field identifiers and aliases. In the following example, the alias "MyState" is used internally to store the data from the database column "STATEID."

```
DIRECT QUERY Dimension STATEID as MyState Measure AMOUNT from SALES_TABLE;
```

This differs from the result of an SQL Select statement with an alias. If the alias is not explicitly quoted, the result contains the default case of column returned by the target database. In the following example, the SQL Select statement to an Oracle database creates "MYSTATE," with all upper case letters, as the internal QlikView alias even though the alias is specified as mixed case. The SQL Select statement uses the column name returned by the database, which in the case of Oracle is all upper case.

```
SQL Select STATEID as MyState, STATENAME from STATE_TABLE;
```

To avoid this behavior, use the LOAD statement to specify the alias.

```
Load STATEID as MyState, STATENAME;
```

```
SQL Select STATEID, STATEMENT from STATE_TABLE;
```

In this example, the "STATEID" column is stored internally by QlikView as "MyState."

Most database scalar expressions are allowed as field specifications. Function calls can also be used in field specifications. Expressions can contain constants that are boolean, numeric, or strings contained in single quotation marks (embedded single quotation marks are represented by adjacent single quotation marks).

Examples:

```
DIRECT QUERY DIMENSION SalesOrderID, RevisionNumber MEASURE SubTotal AS
"Sub Total" FROM AdventureWorks.Sales.SalesOrderHeader
```

```
DIRECT QUERY DIMENSION "SalesOrderID" AS "Sales Order ID" MEASURE Sub-
Total,TaxAmt, (SubTotal-TaxAmt) AS "Net Total" FROM Adven-
tureWorks.Sales.SalesOrderHeader
```

```
DIRECT QUERY DIMENSION (2*Radius*3.14159) AS Circumference, Mole-
cules/6.02e23 AS Moles MEASURE Num1 AS numA FROM TableName
```

```
DIRECT QUERY DIMENSION concat(region, 'code') AS region_code MEASURE Num1
AS NumA FROM TableName
```

Direct Discovery does not support using aggregations in load statements. If aggregations are used, the results are unpredictable. A load statement such as the following should not be used:

```
DIRECT QUERY DIMENSION stateid, SUM(amount*7) AS MultiFirst MEASURE amount
FROM sales_table
```

The SUM should not be in the load statement.

---



Direct Discovery also does not support QlikView functions in Direct Query statements. For example, the following specification for a DIMENSION field results in a failure when the "Mth" field is used as a dimension in a Chart:

```
month(ModifiedDate) as Mth
```

## Native data-source syntax

By design, the DIRECT QUERY statement is data-source neutral for data sources that support SQL. For that reason, the same DIRECT QUERY statement can be used for different SQL databases without change. Direct Discovery generates database-appropriate queries as needed.

Native data-source syntax can be used when the user knows the database to be queried and wants to exploit database-specific extensions to SQL. Native data-source syntax is supported:

- As field expressions in DIMENSION and MEASURE clauses
- As the content of the WHERE clause

Examples:

```
DIRECT QUERY
```

```
    DIMENSION Dim1, Dim2
    MEASURE
        NATIVE ('X % Y') AS X_MOD_Y
    FROM TableName
```

```
DIRECT QUERY
```

```
    DIMENSION Dim1, Dim2
    MEASURE X, Y
    FROM TableName
    WHERE NATIVE ('EMAIL MATCHES "\*.EDU"')
```

## Table names

The database table name specified with the FROM keyword consists of one or more field names, separated by periods. Both simple and quoted names are allowed. Whether or not the table name is case-sensitive depends on the database.

## Where clauses

The full syntax of database WHERE clauses is not defined here, but most SQL "relational expressions" are allowed, including the use of function calls, the LIKE operator for strings, IS NULL and IS NOT NULL, and IN. BETWEEN is not included.

NOT is a unary operator, as opposed to a modifier on certain keywords.

Examples:

```
WHERE x > 100 AND "Region Code" IN ('south', 'west')
WHERE Code IS NOT NULL and Code LIKE '%prospect'
WHERE NOT X in (1,2,3)
```

This last example cannot be written as:

```
WHERE X NOT in (1,2,3)
```

## Restricted Terms

The following terms are used as keywords and so cannot be used as column or field names without being quoted:

- and
- as
- detach
- detail
- dimension
- distinct
- from
- in
- is
- like
- measure
- native
- not
- or
- where

## Direct Discovery System Variables

### DirectCacheSeconds

You can set a caching limit to the Direct Discovery query results for Charts. Once this time limit is reached, QlikView Server clears the cache when new Direct Discovery queries are made. QlikView queries the source data for the selections and creates the cache again for the designated time limit. The result for each combination of selections is cached independently. That is, the cache is refreshed for each selection independently, so one selection refreshes the cache only for the fields selected, and a second selection refreshes cache for its relevant fields. If the second selection includes fields that were refreshed in the first selection, they are not updated in cache again if the caching limit has not been reached.

The Direct Discovery cache does not apply to Table Boxes. Table Box selections query the data source every time.

The limit value must be set in seconds. The default cache limit is 1800 seconds (30 minutes).

The value used for DirectCacheSeconds is the value set at the time the DIRECT QUERY statement is executed. The value cannot be changed at runtime.

```
SET DirectCacheSeconds=1800
```

### DirectConnectionMax

You can do asynchronous, parallel calls to the database by using the connection pooling capability. The load script syntax to set up the pooling capability is as follows:

```
SET DirectConnectionMax=10
```

The numeric setting specifies the maximum number of database connections the Direct Discovery code should use while updating Sheet objects. The default setting is 1.

---

Notes:

Caution should be taken when setting this variable. Setting it to greater than 1 is known to cause problems when connecting to MS SQL Server.

In the earlier release of Direct Discovery, this variable was named "LinkedConnectionMax" and the default was 4.

---

### DirectUnicodeStrings

---

Direct Discovery can support the selection of extended Unicode data by using the SQL standard format for extended character string literals (N'<extended string>') as required by some databases (notably SQL Server). The use of this syntax can be enabled for Direct Discovery with the script variable `DirectUnicodeStrings`.

Setting this variable to 'true' will enable the use of the ANSI standard wide character marker “N” in front of the string literals. Not all databases support this standard. The default setting is 'false'.

### **DirectDistinctSupport**

When a `DIMENSION` field value is selected in a QlikView object, a query is generated for the source database (see *Differences between Direct Discovery and In-Memory Data* (page 165)). When the query requires grouping, Direct Discovery uses the `DISTINCT` keyword to select only unique values. Some databases, however, require the `GROUP BY` keyword. Set `DirectDistinctSupport` to "false" to generate `GROUP BY` instead of `DISTINCT` in queries for unique values.

```
SET DirectDistinctSupport=false
```

If `DirectDistinctSupport` is set to true, then `DISTINCT` is used. If it is not set, the default behavior is to use `DISTINCT`.

## **Direct Discovery Character Variables**

### **DirectFieldColumnDelimiter**

You can set the character used as the field delimiter in Direct Query statements for databases that require a character other than comma as the field delimiter. The specified character must be surrounded by single quotation marks in the SET statement.

```
SET DirectFieldColumnDelimiter= '|'
```

### **DirectStringQuoteChar**

You can specify a character to use to quote strings in a generated query. The default is a single quotation mark. The specified character must be surrounded by single quotation marks in the SET statement.

```
SET DirectStringQuoteChar= '"'
```

### **DirectIdentifierQuoteStyle**

You can specify that non-ANSI quoting of identifiers be used in generated queries. At this time, the only non-ANSI quoting available is GoogleBQ. The default is ANSI. Uppercase, lowercase, and mixed case can be used (ANSI, ansi, Ansi).

```
SET DirectIdentifierQuoteStyle="GoogleBQ"
```

For example, ANSI quoting is used in the following SELECT statement:

```
SELECT [Quarter] FROM [qvTest].[sales] GROUP BY [Quarter]
```

When `DirectIdentifierQuoteStyle` is set to "GoogleBQ", the SELECT statement would use quoting as follows:

```
SELECT [Quarter] FROM [qvTest.sales] GROUP BY [Quarter]
```

### **DirectIdentifierQuoteChar**

You can specify a character to control the quoting of identifiers in a generated query. This can be set to either one character (such as a double quotation mark) or two (such as a pair of square brackets). The default is a double quotation mark.

```
SET DirectIdentifierQuoteChar='YYYY-MM-DD'
```

### **DirectTableBoxListThreshold**

When Direct Discovery fields are used in a Table Box, a threshold is set to limit the number of rows displayed. The default threshold is 1000 records. The default threshold setting can be changed by setting the `DirectTableBoxListThreshold` variable in the load script. For example:

```
SET DirectTableBoxListThreshold=5000
```

The threshold setting applies only to Table Boxes that contain Direct Discovery fields. Table Boxes that contain only in-memory fields are not limited by the `DirectTableBoxListThreshold` setting.

No fields are displayed in the Table Box until the selection has fewer records than the threshold limit.

## Directory

The *Load* (page 253) statement can be preceded by a **directory** statement, defining what directory to look for the files in.

```
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* (page 253) statements, until a new **directory** statement is made.

A **directory** statement is generated automatically if **Relative Paths** is enabled in the **Edit Script** dialog.

Back to *Script Statements and Keywords* (page 219).

## Disconnect

Terminates the current ODBC/OLEDB/Custom 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;
```

Back to *Script Statements and Keywords* (page 219).

## Do..loop

The **do..loop** control statement is a script iteration construct which executes one or several statements 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 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 and as such is ended with either a semicolon or end-of-line, each of its three possible clauses (**do**, **exit do** and **loop**) must not cross a line boundary.

### Examples:

```
// load files file1.csv..file9.csv
Set a=1;
Do while a<10
Load * from file$(a).csv;
Let a=a+1;
Loop
```

Back to *Script Statements and Keywords (page 219)*.

## Drop Field

One or several QlikView fields can be dropped from the data model and thus from memory at any time during script execution, by means of a **drop field** statement.

The syntax is:

```
drop field[ s ] fieldname [ , fieldname2 ... ] [from tablename1 [ , tablename2 ... ]]
```

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;
```

Back to *Script Statements and Keywords (page 219)*.

## Drop Table

One or several QlikView internal tables can be dropped from the data model and thus from memory at any time during script execution, by means of a **drop table** statement. The following items will be lost as a result of this:

- The actual 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 ... ]
```

The forms **drop table** and **drop tables** are both accepted.

### Examples:

```
drop table Orders, Salesmen, T456a;
```

This line results in three tables being dropped from memory.

### Tab1:

```
SQL SELECT * FROM Trans;
LOAD Customer, Sum( sales ) resident Tab1 group by Month;
drop table Tab1;
```

As a result only the aggregates remain in the memory. *Trans data* is discarded.

Back to *Script Statements and Keywords (page 219)*.

## Execute

With the **Execute** statement it is possible to run other programs during the re-execution of the QlikView script, e.g. conversions that are necessary.

The syntax is:

```
execute commandline
```

where:

*commandline* is a text that can be interpreted by the operating system as a command line.

### Examples:

```
Execute C:\Program Files\Office12\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.

---

Back to *Script Statements and Keywords (page 219)*.

## 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 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 and as such is ended with either a semicolon or end-of-line, it must not cross a line boundary.

### Examples:

```
exit script
exit script;
exit script when a=1
```

Back to *Script Statements and Keywords (page 219)*.

## First

The **First** prefix to a *Load (page 253)* or *Select (SQL) (page 270)* statement is used for loading just a set maximum number of records from the data source, 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) SQL SELECT * FROM Orders; (Note: The parenthesis is allowed but not required.)
```

Back to *Script Statements and Keywords (page 219)*.

## For..next

The **for..next** control statement is a script iteration construct 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:

```
forcounter = expr1 to expr2 [ step expr3 ]
```

```
[statements]
```

```
[exit for [ ( when | unless ) condition ]
```

```
[statements]
```

```
next [counter]
```

Where:

*counter* is a 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 determines the first value of the *counter* variable for which the loop should be executed.

*expr2* is an expression which determines the last value of the *counter* variable for which the loop should be executed.

*expr3* is an expression which determines the value indicating the increment of the *counter* variable each time the loop has been executed.

*condition* is a 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 this is not good programming practice.

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 and as such is ended with either a semicolon or end-of-line, each of its three possible clauses (**for..to..step**, **exit for** and **next**) must not cross a line boundary.

### 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
```

Back to *Script Statements and Keywords* (page 219).

## For each..next

The **for each..next** control statement is a script iteration construct which executes one or several statements 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 each 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) | filelistmask | dirlistmask
```

*constant* is any number or string. Note that a string written directly in the script must be enclosed by single quotes. A string without single quotes will be interpreted as a variable and then the value of the variable will be used. Numbers need not be enclosed by single quotes.

*expression* is an arbitrary expression.

*mask* is a filename or directory name mask which may include any valid filename characters as well as the standard wildcard characters, \* and ?.

*condition* is a 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 this is not good programming practice.

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 and as such is ended with either a semicolon or end-of-line, each of its three possible clauses (**for each**, **exit for** and **next**) must not cross a line boundary.

### 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
ENDSUB
CALL DoDir ('C:')

```

Back to *Script Statements and Keywords (page 219)*.

## Force

The **force** statement forces QlikView to interpret field values of subsequent *Load (page 253)* and *Select (SQL) (page 270)* statements as written with only upper case letters, with only lower case letters, as always capitalized or as they appear (mixed). This statement makes it possible to associate field values from 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.

Back to *Script Statements and Keywords (page 219)*.

## Generic

The unpacking and loading of a generic database can be done with a **generic** prefix. For a description of a generic database, see chapter *Generic Database* in section *Evaluating the Loaded Data (page 391)*.

Tables loaded through a generic statement are not auto-concatenated.

The syntax is:

```
Generic ( loadstatement | selectstatement )
```

### Examples:

```
Generic LOAD * FROM abc.csv;  
Generic SQL SELECT * FROM table1;
```

Back to *Script Statements and Keywords (page 219)*.

## Hierarchy

The **hierarchy** prefix is used to transform a hierarchy table to a table that is useful in a QlikView data model.

It can be put in front of a **Load** or a **Select** statement and will use the result of the loading statement as input for a table transformation.

The input table must be an adjacent nodes table.

Adjacent nodes tables are tables where each record corresponds to a node and has a field that contains a reference to the parent node. In such a table the node is stored on one record only but the node can still have any number of children. The table may of course contain additional fields describing attributes for the nodes.

The prefix creates an expanded nodes table, which normally has the same number of records as the input table, but in addition 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 and the path field can be used in a tree structure. More details about adjacent nodes tables and expanded nodes tables can be found in the chapter *Hierarchies* in section *Evaluating the Loaded Data (page 391)*.

Usually 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 in the input table. If so, the output table may have more records than the input table.

All nodes with a parent id not found in the **nodeid** column (including nodes with missing parent id) 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 | select-  
statement)
```

where

**NodeID** is the name of the field that contains the node id. This field must exist in the input table.

**ParentID** is the name of the field that contains the node id of the parent node. This field must exist in the input table.

**NodeName** is the name of the field that contains the name of the node. This field must exist in the input table.

**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$];
```

Back to *Script Statements and Keywords (page 219)*.

## HierarchyBelongsTo

The **hierarchybelongsto** prefix is used to transform a hierarchy table to a table that is useful in a QlikView data model.

It can be put in front of a **Load** or a **Select** statement and will use the result of the loading statement as input for a table transformation.

The input table must be an adjacent nodes table.

Adjacent nodes tables are tables where each record corresponds to a node and has a field that contains a reference to the parent node. In such a table the node is stored on one record only but the node can still have any number of children. The table may of course contain additional fields describing attributes for the nodes. More details about adjacent nodes tables and expanded nodes tables can be found in the chapter *Hierarchies* in section *Evaluating the Loaded Data (page 391)*.

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 in most cases 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. This field must exist in the input table.

**ParentID** is the name of the field that contains the node id of the parent node. This field must exist in the input table.

**NodeName** is the name of the field that contains the name of the node. This field must exist in the input table.

**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$]);
```

Back to *Script Statements and Keywords (page 219)*.

## If..then..elseif..else..end if

The **if..then** control statement is a script selection construct forcing the script execution to follow different paths depending on one or several 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 and as such is ended with either a semicolon or end-of-line, each of its four possible clauses (**if..then**, **elseif..then**, **else** and **end if**) must not cross a line boundary.

**Examples:**

```
if a=1 then
load * from abc.csv;
sql select e, f, g from tab1;
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.txt;
end if
```

Back to *Script Statements and Keywords (page 219)*.

## Image\_size

This clause is used with the *Info (page 245)* prefix to resize images from a database management system to fit in the fields. Width and height are specified in pixels.

### Example:

```
Info Image_size(122,122)Select ID, Photo From infotable;
```

Back to *Script Statements and Keywords (page 219)*.

## Info

If a piece of external information, such as a text file, a picture or a video is to be linked to a field value, this is done in a table that is loaded using an **info** prefix. (In some cases it will be preferable to store the information inside the .qvw file, by using the *Bundle (page 224)* prefix.) The table must contain two columns only, the first one with the field values that will form the keys to the information, the second one containing the information elements, e.g. the file names of the pictures etcetera.

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 SQL SELECT * FROM infotable;
Info SQL SELECT Key, Picture FROM infotable;
```

Back to *Script Statements and Keywords (page 219)*.

## Inner

The *Join (page 250)* and *Keep (page 250)* prefixes can be preceded by the prefix **inner**.

If used before **join** it specifies that an inner join should be used. The resulting table will thus only contain combinations of field values from the raw data tables where the linking field values are represented in both tables.

If used before **keep**, it specifies that both raw data tables should be reduced to their common intersection before being stored in QlikView.

```
inner ( join | keep ) [ (tablename ) ](loadstatement |selectstatement )
```

### Examples:

#### Table1

A	B
1	aa
2	cc

```
3 ee
```

**Table2**

```
A C
```

```
1 xx
```

```
4 yy
```

```
QVTable:
```

```
SQL SELECT * FROM table1;
```

```
inner join SQL SELECT * FROM table2;
```

**QVTable**

```
A B C
```

```
1 aa xx
```

```
QVTab1:
```

```
SQL SELECT * FROM Table1;
```

```
QVTab2:
```

```
inner keep SQL 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 A.

Back to *Script Statements and Keywords* (page 219).

## 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** statements.

Input fields behave somewhat differently from 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. The replacement value is user dependent, i.e. when an input field is used on a server, different users will see different sets of input field values

In input fields all field values will be treated as distinct, regardless if several have the same value.

The syntax is:

```
inputfield fieldlist
```

*fieldlist* is a comma separated list of the fields that should be flagged as input fields. The wildcard characters \* and ? are allowed in field names. Quoting of field names may be necessary when wildcards are used.

### Examples:

```
Inputfield B;
Inputfield A,B;
Inputfield B??x*;
```

Back to *Script Statements and Keywords (page 219)*.

## IntervalMatch

The **IntervalMatch** prefix is used to create a table matching discrete numeric values to one or more numeric intervals.

It must be placed before a *Load (page 253)* or *Select (SQL) (page 270)* statement that loads the intervals. The field containing the discrete data points (Time in the example below) must already have been loaded into QlikView before the statement with the IntervalMatch prefix. The prefix does not by itself read this field from the database table. The prefix transforms the loaded table of intervals to a table that contains an additional column: the discrete numeric data points. It also expands the number of records so that the new table has one record per possible combination of discrete data point and interval.

The intervals may be overlapping and the discrete values will be linked to all matching intervals.

The general syntax is:

```
intervalmatch (matchfield) (loadstatement | selectstatement )
```

*matchfield* is the field containing the discrete numeric values to be linked to intervals.

*loadstatement* or *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 (undefined).

There is also an extended syntax of IntervalMatch including one or several additional key fields. See *IntervalMatch (Extended Syntax) (page 249)*.

### Example:

In the two tables below, the first one defines the start and end times for the production of different orders. The second one lists a number of discrete events. By means of the IntervalMatch prefix it is possible to logically connect the two tables in order to find out e.g. which orders were affected by 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, then link the field *Time* to the time intervals defined by the fields *Start* and *End*:

```
OrderLog:
LOAD * INLINE [
  Start, End, Order
  01:00, 03:35, A
  02:30, 07:58, B
  03:04, 10:27, C
  07:23, 11:43, D
];
EventLog:
LOAD * INLINE [
  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
];
IntervalMatch (Time) LOAD Start, End Resident OrderLog;
```

The following table box can now be created in QlikView:



Tablebox

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

Back to *Script Statements and Keywords (page 219)*.

## IntervalMatch (Extended Syntax)

The extended **IntervalMatch** prefix is used to create a table matching discrete numeric values to one or more numeric intervals, while at the same time matching the values of one or several additional keys.

This is a very powerful and flexible feature that can be used for linking transactions with dimensions that are changing over time: Slowly changing dimensions.

The IntervalMatch prefix must be placed before a *Load (page 253)* or *Select (SQL) (page 270)* statement that loads the intervals. The table containing the discrete data points and the additional keys must already have been loaded into QlikView before the statement with the IntervalMatch prefix. The prefix transforms the loaded table of intervals and keys to a table that contains an additional column: the discrete numeric data points. It also expands the number of records so that the new table has one record per possible combination of discrete data point, interval and value of the key field(s).

The syntax is:

```
intervalmatch (matchfield,keyfield1 [ , keyfield2, ... keyfield5 ]
) (loadstatement | selectstatement )
```

*matchfield* is the field containing the discrete numeric values to be linked to intervals.

*keyfield(s)* are fields that contain the additional attributes that are to be matched in the transformation.

*loadstatement* or *selectstatement* must result in a table where the first two fields contain the lower and upper limits of each interval and the third and any subsequent fields contain the *keyfield(s)* present in the **IntervalMatch** statement. The intervals are always closed, i.e. the end points are included in the interval. Non-numeric limits render the interval to be disregarded (undefined).

In order to avoid undefined interval limits being disregarded, it may be necessary to allow NULL values to map to other fields that constitute the lower or upper limits to the interval. This can be handled by the **NullAsValue (page 262)** statement or by an explicit test that replaces NULLs with a numeric value well before or after any of the discrete numeric data points.

### Example:

```
Inner Join IntervalMatch (Date,Key) LOAD FirstDate, LastDate, Key res-
ident Key;
```

Back to *Script Statements and Keywords (page 219)*.

## Join

The **join** prefix joins the loaded table with an existing named table or the last previously created data table. The join is a *Natural Join* made over all the common *Fields (page 171)*. The join prefix may be preceded by one of the prefixes *Inner (page 245)*, *Outer (page 263)*, *Left (page 251)* or *Right (page 267)*.

The syntax is:

```
[inner | outer | left | right ]join [ (tablename ) ]( loadstatement
| selectstatement )
```

### Examples:

```
Join LOAD * FROM abc.csv;

Join SQL SELECT * FROM table1;

tab1:
LOAD * FROM file1.csv;
tab2:
LOAD * FROM file2.csv;
.. .. .
join (tab1) LOAD * FROM file3.csv;
```

Back to *Script Statements and Keywords (page 219)*.

## Keep

The **keep** prefix is similar to the *Join (page 250)* prefix. Just as the join prefix, it compares the loaded table with an existing named table or the last previously created data table.

But instead of joining the loaded table with an existing table, it 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 comparison made is equivalent to a natural join made over all the common fields, i.e. the same way as in a corresponding join. However, the two tables are not joined and will be kept in QlikView as two separately named tables.

---

### Note!

The explicit *Join (page 250)* prefix in 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 multiple tables instead of joining them, which greatly reduces memory usage, increases processing speed and offers enormous flexibility. Explicit joins should therefore generally be avoided in QlikView scripts. The **keep** functionality was designed to reduce the number of cases where explicit joins needs to be used.

---

The keep prefix must be preceded by one of the prefixes *Inner (page 245)*, *Left (page 251)* or *Right (page 267)*. The syntax is:

```
(inner | left | right) keep[ (tablename ) ]( loadstatement | select-
statement )
```

### Example:

```

Inner Keep LOAD * FROM abc.csv;
Left Keep SQL SELECT * FROM table1;
tab1:
LOAD * FROM file1.csv;
tab2:
LOAD * FROM file2.csv;
... ..
Left keep (tab1) LOAD * FROM file3.csv;

```

Back to *Script Statements and Keywords (page 219)*.

## Left

The *Join (page 250)* and *Keep (page 250)* prefixes can be preceded by the prefix **left**.

If used before **join** it specifies that a left join should be used. The resulting table will only contain combinations of field values from the raw data tables where the linking field values are represented in the first table.

If used before **keep**, it specifies that the second raw data table should be reduced to its common intersection with the first table, before being stored in QlikView.

```
left ( join | keep ) [ (tablename ) ] (loadstatement | selectstatement )
```

### Examples:

#### 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;

```

A	B	C
1	aa	xx
2	cc	
3	ee	

```
QVTab1:
select * from Table1;
QVTab2:
left keep select * from Table2;
```

A	B
1	aa
2	cc
3	ee

A	C
1	xx

The two tables in the **keep** example are, of course, associated via A.

```
tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
. . . .
left keep (tab1) load * from file3.csv;
```

Back to *Script Statements and Keywords* (page 219).

## Let

The **let** statement has been created as a complement to the *Set* (page 272) statement, used for defining *Script Variables* (page 282). The **let** statement, in opposition to the **set** statement, evaluates the expression on the right side of the '=' before it is assigned to the 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 end-of-line.

The syntax is:

```
let variablename=expression
```

Note that the word **let** may be omitted.

```
Set x=3+4;
Let y=3+4;
z=$(y)+1;
```

$\$(x)$  will be evaluated as '3+4'

$\$(y)$  will be evaluated as '7'

$\$(z)$  will be evaluated as '8'

### Example:

```
Let T=now( );
```

$\$(T)$  will be given the value of the current time.

Back to *Script Statements and Keywords (page 219)*.

## Load

The **load** statement loads fields from a file, from data defined in the script, from a previously loaded table, from a web page, from the result of a subsequent **select** statement or by generating data automatically. The general syntax of the load statement is:

```
load [ distinct ] *fieldlist
[ ( from file [ format-spec ] |
from_field fieldsource [ format-spec ]
inline data [ format-spec ] |
resident table-label |
autogenerate size ) ]
[ where criterion | while criterion ]
[ group_by groupbyfieldlist ]
[ order_by orderbyfieldlist ]
```

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 loaded. Using \* as field list indicates all fields in the table.

*field* ::= ( *fieldref* | *expression* ) [ **as** *aliasname* ]

The field definition must always contain a literal, 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 table 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 preceded 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 BCD (Binary Coded Decimal) numbers 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 further information, see the syntax of *Script Expressions* (page 295).

**as** is used for assigning a new name to the field.

---

**Note!**

If a table contains two fields with the same field name, typically when loading from a text file, only one field can be loaded even though the field names have been changed using **as**.

---

**from** is used if data should be loaded from a file.

*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* (page 257), within brackets.

**from\_field** is used if data should be loaded from a previously loaded field.

*fieldsource ::= (tablename, fieldname)*

The field is the name of the previously loaded *tablename* and *fieldname*.

*format-spec ::= ( fspec-item { , fspec-item } )*

The format specification consists of a list of several *Format Specification Items* (page 257), within brackets.

**inline** is used if data should be typed within the script, and not loaded from a file. Use the *Inline Data Wizard* (page 194) for help with the creation of **load inline** statements.

*data ::= [ text ]*

Data entered through an **inline** clause must be enclosed by double *Quotation Marks in Scripting* (page 292) or with 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 table.

*table label* is a label preceding the **load** or **select** statement(s) that created the original table. The label should 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 which require data from a database. Only constants and parameter-free functions (e.g. rand(), recno()) are allowed in the expressions.

**where** is a clause used for stating whether a record should be included in the selection or not. The selection is included if *criterion* is true.

---

**while** is a clause used for stating 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.

*criterion* is a logical expression.

**group by** is a clause used for defining over which fields the data should be aggregated (grouped). The aggregation fields should be included in some way in the expressions loaded. No other fields than the aggregation fields may be used outside aggregation functions in the loaded expressions.

*groupbyfieldlist ::= (fieldname { .fieldname } )*

*order by* is a clause used for sorting the records of a resident 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. The ordering fields specify which fields the resident table is sorted by. The field can be specified by its name or by its number in the resident table (the first field is number 1).

*orderbyfieldlist ::= fieldname [ sortorder ] { , fieldname [ sortorder ] }*

*sortorder* is either asc for ascending or desc for descending. If no *sortorder* is specified, asc is assumed.

*fieldname*, *path*, *filename* and *aliasname* are text strings representing what the respective names imply. Any field in the source table can be used as *fieldname*. However, fields created through the as clause (*aliasname*) are out of scope and cannot be used inside the same load statement.

---

#### Note!

If no source of data is given by means of a from, inline, resident, from field or autogenerate clause, data will be loaded from the result of the immediately succeeding select or load statement. The succeeding statement should not have a prefix.

---

#### 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);
Load * from C:\qdssamples\xyz.qvx (qvx);
```

*Selecting certain fields, calculating fields*

```
Load FirstName, LastName, Number from data1.csv;
Load @1as 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;
```

*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); (Note: The parenthesis  
after autogenerate is allowed but not required.)
```

#### *Loading data from previously loaded table*

```
tab1:  
Select A,B,C,D from transtable;  
Load A,B,month(C),A*B+D as E resident tab1;  
Load A,A+B+C resident tab1 where A>B;  
Load A,B*C as E resident tab1 order by A;  
Load A,B*C as E resident tab1 order by 1,2;  
Load A,B*C as E resident tab1 order by C desc, B asc, 1 desc;
```

#### *Loading data from previously loaded fields*

```
Load A from _field (Characters, Types);
```

#### *Loading data from succeeding table*

```
Load A, B, if(C>0,'positive','negative') 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)<>' ';
```

Back to *Script Statements and Keywords (page 219)*.



## Format Specification Items

Each format specification item defines a certain property of the table file:

```
fspec-item ::= [ ansi | oem | mac | UTF-8 | Unicode | txt | fix | dif | biff | ooxml | html | xml | qvd | 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 | record is n | record is line | record is n lines | no quotes | msq | filters (filter specifiers ) ]
```

The following properties can be defined:

<b>Character Set</b>	<i>page 407</i>
<b>Table Format</b>	<i>page 257</i>
<b>Delimiter</b>	<i>page 258</i>
<b>No eof</b>	<i>page 258</i>
<b>Labels</b>	<i>page 259</i>
<b>Header Size</b>	<i>page 259</i>
<b>Record Length</b>	<i>page 259</i>
<b>Quotes</b>	<i>page 260</i>

## Character Set

Character set is a file specifier for the **Load** statement that defines the character set used in the file.

Available character sets are:

```
ansi
oem
mac
utf8
unicode
codepage is N
```

The file can be written with the **ansi** character set (Windows), with the **oem** character set (DOS, OS/2, AS400 and others), **unicode**, **utf8** or with the **mac**. The conversion from the **oem** character set is not implemented for MacOS. With the codepage specifier, it is possible to use any Windows codepage. If nothing is specified, codepage 1252 is assumed under Windows.

The **ansi**, **oem** and **mac** specifiers were used in older versions of QlikView and will still work. However, they will not be generated when creating the **Load** statement with a modern QlikView.

### Examples:

```
LOAD * FROM a.txt (utf8, txt, delimiter is ',', embedded labels)
```

```
LOAD * FROM a.txt (unicode, txt, delimiter is ',', embedded labels)
```

```
LOAD * FROM a.txt (codepage is 10000, txt, delimiter is ',', no labels)
```

Back to *Script Statements and Keywords (page 219)*.

## Table Format

The table format is a file specification for the **Load** statement that defines which file type it is.

<b>txt</b>	In a delimited text file, <b>.txt</b> , the columns in the table are separated by some character.
<b>fix</b>	In a fixed record length file, <b>.fix</b> , each column is exactly a certain number of characters wide.
<b>dif</b>	In a <b>.dif</b> file, (Data Interchange Format) a special format for defining the table is used.
<b>biff</b>	QlikView can also interpret data in standard Excel files by means of the <b>biff</b> format (Binary Interchange File Format).
<b>ooxml</b>	QlikView can interpret data in the Excel 2007 <b>ooxml</b> format (OpenOfficeXML).
<b>html</b>	If the table is part of an html page or file, <b>html</b> should be used.
<b>qvd</b>	The format <b>qvd</b> is the proprietary <i>QVD Files (page 409)</i> format, exported from a QlikView document.
<b>qvx</b>	<b>qvx</b> is a file/stream format for high performance output to QlikView.

If nothing is specified, a **.txt** file is assumed.

Back to *Script Statements and Keywords (page 219)*.

## Delimiter

This specifier is relevant only for delimited **.txt** files.

```
delimiter is char
```

For delimited table files, an arbitrary delimiter (from the 127 ASCII characters) can be specified through the **delimiter is** specifier. *char* specifies a single character.

There are some special cases:

"\t" representing a tab sign, with or without quotation marks.

"\" representing a backslash ( \ ) character.

the word "**spaces**" representing all combinations of one or more spaces. Non-printable characters with an ASCII-value below 32, with the exception of CR and LF, will be interpreted as spaces.

If nothing is specified, **delimiter is ','** is assumed.

Back to *Script Statements and Keywords (page 219)*.

## No eof

This specifier is relevant only for delimited **.txt** files.

```
no eof
```

If this option is used, ASCII character 26 which otherwise denotes end-of-file is disregarded. If this specifier is used, character 26 can be part of a field value.

Back to *Script Statements and Keywords (page 219)*.

## Labels

Labels is a file specifier for the *Load* (page 253) statement that defines where in a file the field names can be found.

The possible alternatives are:

**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.

### Examples:

```
LOAD * FROM a.txt (unicode, txt, delimiter is ',' , embedded labels)
LOAD * FROM a.txt LOAD * FROM a.txt (unicode, txt, delimiter is ',' ,
embedded labels)
```

Back to *Script Statements and Keywords* (page 219).

## Header Size

Specifies the header size in table files.

The syntax is:

**header is *n***

**header is line**

**header is *n* lines**

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* must be a positive integer, representing the header length. If not specified, **header is 0** is assumed. The **header is** specifier is only relevant for Table file.

Back to *Script Statements and Keywords* (page 219).

## Record Length

Specifies the record length for fixed record files.

**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**), where *n* is a positive integer representing the record length. The **record is** specifier is only relevant for **fix** files.

Back to *Script Statements and Keywords* (page 219).

## Quotes

**Quotes** is a file specifier for the **Load** statement that defines whether quotes can be used and the precedence between quotes and separators.

For text files only. Available options are:

```
no quotes
```

```
msq
```

If the specifier is omitted, standard quoting is used, i.e. the quotes " " or ' ' can be used, but only if they are the first and last non blank character of a field value.

If *Quotation Marks in Scripting* (page 292) are not to be accepted in a text file, the **no quotes** specifier should be used.

**msq** is used to specify modern style quoting, allowing multi-line content in fields. Fields containing end-of-line characters must be enclosed within double quotes.

---

One limitation of the **msq** specifier is that single double-quote (") characters appearing as first or last character in field content will be interpreted as start or end of multi-line content, which may lead to unpredicted results in the data set loaded. In this case you should use standard quoting instead, omitting the specifier.

---

Back to *Script Statements and Keywords* (page 219).

## 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, the path must be quoted.

---

### 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.

---

Back to *Script Statements and Keywords* (page 219).

## Loosen Table

When data that includes circular references is loaded into QlikView, loosely coupled tables are created automatically, to avoid that the circular references create a loop in the QlikView internal logic.

One or more QlikView internal data tables can be explicitly declared loosely coupled during script execution by using a **Loosen Table** statement.

The use of one or more **Loosen Table** statements in the script will make QlikView disregard any setting of tables as loosely coupled made before the script execution.

The syntax is:

```
Loosen Table[s] tablename [ , tablename2 ...]
```

Either syntax: **Loosen Table** and **Loosen Tables** can be used.

### Example:

```
Tab1:
```

---

```
Select * from Trans;
Loosen table Tab1;
```

**Note!**

Should QlikView find 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 tables will be forced loosely coupled until no circular references remain. When this happens, the **Loop Warning** dialog, gives a warning.

**See also:**

*Circular References (page 111)*

*Intentionally Creating Loosely Coupled Tables (page 925)*

Back to *Script Statements and Keywords (page 219)*.

## Mapping

The **mapping** prefix is used to create a mapping table that can be used for e.g. replacing field values and field names during the script run.

It can be put in front of a *Load (page 253)* or *Select (SQL) (page 270)* statement and will store the result of the loading statement as a mapping table. A mapping table consists of two columns, the first containing comparison values and the second containing the desired mapping values. Mapping tables are stored temporarily in the memory and dropped automatically after the script execution.

The content of the mapping table can be accessed using e.g. the **map ... using** statement, the **rename field** statement, the **Applymap()** function or the **Mapsubstring()** function.

The syntax is:

```
mapping ( loadstatement | selectstatement )
```

**Examples:**

```
Mapping LOAD * FROM x.csv
mapping SQL SELECT a, b FROM map1
map1:
mapping LOAD * inline [
x,y
US,USA
U.S.,USA
America,USA ];
```

Back to *Script Statements and Keywords (page 219)*.

## Map ... using

The **map ... using** statement is used for mapping a certain field value or expression to the values of a specific mapping table. The mapping table is created through the *Mapping (page 261)* statement.

The automatic mapping is done for fields loaded after the **map ... using** statement until the end of the script or until an **unmap** statement is encountered.

The mapping is done last in the chain of events leading up to the field being stored in the internal table in QlikView. This means that 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 internal table. If mapping on the expression level is required, the **Applymap()** function has to be used instead.

**Note!**

The **map ... using** statement does not work when loading QVD files in optimized Mode.

---

The syntax is:

```
map *fieldlist using mapname
```

*\*fieldlist* is a comma separated list of the fields that should be mapped from this point in the script. Using **\*** as field list indicates all fields. The wildcard characters **\*** and **?** are allowed in field names. Quoting of field names may be necessary when wildcards are used.

*mapname* is the name of a mapping table previously read in a **mapping load** or **mapping select** statement.

**Examples:**

<b>map Country using Cmap;</b>	Enables mapping of field Country using map Cmap.
<b>map A, B, C using X;</b>	Enables mapping of fields A, B and C using map X.
<b>map * using GenMap;</b>	Enables mapping of all fields using GenMap.

Back to *Script Statements and Keywords (page 219)*.

## NoConcatenate

The **NoConcatenate** prefix forces two loaded tables with identical field sets to be treated as two separate internal tables, when they otherwise would be automatically be concatenated.

The syntax is:

```
noconcatenate ( loadstatement | selectstatement )
```

**Example:**

```
Load A,B from file1.csv;  
noconcatenate load A,B from file2.csv;
```

Back to *Script Statements and Keywords (page 219)*.

## NullAsValue

The **NullAsValue** statement specifies for which fields the encountered NULLs should be converted to values.

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.

The **NullAsValue** statement operates as a switch and will operate on subsequent loading statements. It can be switched off again by means of the *NullAsNull (page 263)* statement.

The **NullAsValue** can be combined with the variable **NullValue** that specifies which string to use as NULL value. If the variable **NullValue** is not used, **NullAsValue** will replace NULLs with empty strings.

The syntax is:

```
NullAsValue *fieldlist
```

*\*fieldlist* is a comma separated list of the fields for which **NullAsValue** should be turned on. Using **\*** as field list indicates all fields. The wildcard characters **\*** and **?** are allowed in field names. Quoting of field names may be necessary when wildcards are used.

---

**Example:**

```
NullAsValue A,B;
Set NullValue = 'NULL';
Load A,B from x.csv;
```

Back to *Script Statements and Keywords (page 219)*.

**NullAsNull**

The **NullAsNull** statement turns off the conversion of NULLs to string values previously set by a *NullAsValue (page 262)* statement.

The **NullAsValue** statement operates as a switch and can be turned on or off several times in the script, using either a **NullAsValue** or a **NullAsNull** statement.

The syntax is:

```
NullAsNull *fieldlist
```

*\*fieldlist* is a comma separated list of the fields for which **NullAsValue** should be turned off. Using \* as field list indicates all fields. The wildcard characters \* and ? are allowed in field names. Quoting of field names may be necessary when wildcards are used.

**Examples:**

```
NullAsNull A,B;
Load A,B from x.csv;
```

Back to *Script Statements and Keywords (page 219)*.

**Outer**

The explicit *Join (page 250)* prefix 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 resulting table will thus contain combinations of field values from the raw data tables where the linking field values are represented in one or both tables. The **outer** keyword is optional.

```
outer join [ (tablename ) ](loadstatement |selectstatement )
```

**Example:****Table1**

A	B
1	aa
2	cc
3	ee

**Table2**

A	C
---	---

```
1    xx
```

```
4    yy
```

```
select * from table1;
join select * from table2;
      OR
select * from table1;
outer join select * from table2;
```

**Joined table**

A	B	C
1	aa	xx
2	cc	-
3	ee	-
4	-	yy

Back to *Script Statements and Keywords* (page 219).

## Qualify

The **Qualify** statement is used for switching on the qualification of field names, i.e. field names will get the table name as a prefix.

The automatic association of 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.

The qualification will be made for all fields loaded after the **Qualify** statement.

---

**Note!**

The **qualify** statement should not be used in conjunction with partial reload!

---

Qualification is always turned off by default at the beginning of script execution. Qualification of a field name can be activated at any time using a **qualify** statement. Qualification can be turned off at any time using an *Unqualify* (page 280) statement.

The syntax is:

```
qualify *fieldlist
```

*\*fieldlist* is a comma separated list of the fields for which qualification should be turned on. Using **\*** as field list indicates all fields. The wildcard characters **\*** and **?** are allowed in field names. Quoting of field names may be necessary when wildcards are used.

**Examples:**

```
Qualify B;
Load A,B from x.csv;
```



```
Load A,B from y.csv;
```

The two tables **x.csv** and **y.csv** are associated only through **A**. Three fields will result: A, x.B, y.B.

In an unfamiliar database, it is often useful to start out by making sure that only one or a few fields are associated, as illustrated in this example:

```
qualify *;
unqualify TransID;
select * from tab1;
select * from tab2;
select * from tab3;
```

Only **TransID** will be used for associations between the tables *tab1*, *tab2* and *tab3*.

Back to *Script Statements and Keywords (page 219)*.

## Rem

The **rem** statement is used for inserting remarks into the script, or to temporarily deactivate script statements without removing them.

Everything between the **rem** and the next semicolon ; is considered to be a comment.

The syntax is:

```
rem string
```

where:

*string* is an arbitrary text.

### Example:

```
Rem ** This is a comment **;
```

## Tip!

There are two alternative methods available for making comments in the script:

1. It is possible to create a comment anywhere in the script - except between two quotes - by placing the section in question between `/*` and `*/`.  
The commented lines can extend over several subsequent tabs of the script editor. However only the commented lines located on the first tab turn into green color.
2. When typing `//` in the script, all text that follows to the right on the same row becomes a comment. (Note the exception `//:` that may be used as part of an Internet address.)

Back to *Script Statements and Keywords (page 219)*.

## Rename Field

Renames one or more existing QlikView field(s) after they have been loaded.

Two differently named fields cannot be renamed to having the same name. The script will run without errors, but the second field will not be renamed.

The syntax is:

```
rename field (using mapname | oldname to newname{ , oldname to newname }
```

```
rename fields (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 .

*newname* is the new field name.

Either syntax: **rename field** or **rename fields** can be used.

### Examples:

```
Rename field XAZ0007 to Sales;
```

```
FieldMap:
```

```
Mapping SQL SELECT oldnames, newnames FROM datadictionary;
```

```
Rename fields using FieldMap;
```

Back to *Script Statements and Keywords (page 219)*.

## Rename Table

Renames one or more existing QlikView internal table(s) after they have been loaded.

Two differently named tables cannot be renamed to having the same name. The script will run without errors, but the second table will not be renamed.

The syntax is:

```
rename table (using mapname | oldname to newname{ , oldname to newname })
```

```
rename tables (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.

Either syntax: **rename table** or **rename tables** can be used.

### Examples:

```
Tab1:
```

```
Select * from Trans;
```

```
Rename table Tab1 to Xyz;
```

```
TabMap:
```

```
Mapping load oldnames, newnames from tabnames.csv;
```

```
Rename tables using TabMap;
```

Back to *Script Statements and Keywords (page 219)*.

## Right

The *Join* (page 250) and *Keep* (page 250) prefixes can be preceded by the prefix **right**.

If used before **join** it specifies that a right join should be used. The resulting table will only contain combinations of field values from the raw data tables where the linking field values are represented in the second table.

If used before **keep**, it specifies that the first raw data table should be reduced to its common intersection with the second table, before being stored in QlikView.

```
right (join | keep) [(tablename )] (loadstatement | selectstatement )
```

### Examples:

#### 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 A.

```
tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
... ..
right keep (tab1) load * from file3.csv;
```

Back to *Script Statements and Keywords* (page 219).

## Replace

The **replace** prefix can be added to any *Load* (page 253), *Select (SQL)* (page 270) or *Map ... using* (page 261) statement in the script. The **replace load/replace select** statement has the effect of dropping the entire QlikView table, for which a 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 reload* and full reload. The **replace map...using** statement causes mapping to take place also during partial script execution.

The syntax is:

```
replace [only] (loadstatement | selectstatement | map...usingstatement)
```

where:

**only** is an optional qualifier denoting 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 initially dropped. Thereafter 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 initially dropped. Thereafter 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 by new data loaded from File2.csv. All data from File1.csv is lost.

During partial reload, the entire QlikView table Tab1 is initially dropped. Thereafter it is replaced by 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 initially dropped. Thereafter it is replaced by new data loaded from File2.csv. All data from File1.csv is lost.

Back to *Script Statements and Keywords (page 219)*.

## Sample

The **sample** prefix to a *Load (page 253)* or *Select (SQL) (page 270)* statement is used for loading a random sample of records from the data source.

All records will be read but only some of them will be loaded into QlikView.

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.

---

### Note:

It is not possible to use the **sample** prefix when loading an optimized QVD file.

---

### Examples:

```
Sample 0.15 Select * from Longtable;
Sample(0.15) Load * from Longtab.csv; (Note: The parenthesis is
allowed but not required.)
```

Back to *Script Statements and Keywords (page 219)*.

## Section

With the **section** statement, it is possible to define whether the subsequent *Load (page 253)* and *Select (SQL) (page 270)* statements should be considered as data or as a definition of the access rights.

The syntax is:

```
section (access | application)
```

---

**Examples:**

```
Section access;
```

AuthorizationTable:

```
SQL SELECT ACCESS, NTNAME, REGION FROM AuthorizationTable ;
```

```
Section application;
```

If nothing is specified, **section application** is assumed. The **section** definition is valid until a new section statement is made.

---

It is not possible to use the **section access** statement in conjunction with the *Semantic (page 272)* prefix.

---

Back to *Script Statements and Keywords (page 219)*.

**Select (SQL)**

The selection of *Fields (page 171)* from an ODBC data source or from an OLE DB provider is made through standard SQL **SELECT** statements. However, the extent to which the **SELECT** statements are accepted depends largely on the ODBC driver or OLE DB provider used. Here follows 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 = field-
ref ]
```

**distinct** is a predicate used if duplicate combinations of values in the selected fields only should be loaded once.

**distinctrow** is a predicate used if duplicate records in the source table only should be loaded once.

*\*fieldlist* ::= ( \*| *field* ) { ,*field* }

A list of the fields to be selected. Using \* as field list indicates all 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 the ODBC driver or OLE DB provider 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 for assigning 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 for stating 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 the ODBC driver or OLE DB provider for more information.

**group by** is a clause used for aggregating (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 for qualifying groups in a similar manner to how the **where** clause is used for qualifying records.

**order by** is a clause used for stating 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 blank 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 or OLE DB provider specified in the data source definition of the data source in the *Connect (page 228)* 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 might 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** is 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`;
```

Back to *Script Statements and Keywords (page 219)*.

---

## Semantic

Tables containing relations between objects can be loaded through a **semantic** prefix.

The syntax is:

```
semantic ( loadstatement | selectstatement )
```

### Examples:

```
Semantic Load * from abc.csv;  
Semantic Select Object1, Relation, Object2, InverseRelation from  
table1;
```

---

Tables loaded through a **semantic** statement cannot be concatenated.

---

It is not possible to use the **semantic** prefix in conjunction with the *Section (page 269)* statement.

---

Back to *Script Statements and Keywords (page 219)*.

## Set

The **set** statement is used for defining *Script Variables (page 282)*. These can be used for substituting strings, paths, drives, etc.

The syntax is:

```
set variablename=string
```

### Examples:

```
Set FileToUse=Data1.csv;  
Set Constant="My string";  
Set BudgetYear=1997;
```

Back to *Script Statements and Keywords (page 219)*.

## Sleep

The **sleep** statement pauses script execution for  $n$  milliseconds, where  $n$  is a positive integer no larger than 3600000 (i.e. 1 hour). The value may be an expression.

The syntax is:

```
sleep n
```

### Examples:

```
sleep 10000;  
sleep t*1000;
```

Back to *Script Statements and Keywords (page 219)*.

## SQL

The **SQL** statement allows you to send an arbitrary SQL command through an *ODBC* or *OLE DB* connection. Simply type:

```
SQL
```

followed by the command.

---



**Example:**

```
SQL leave;
```

SQL Execute <storedProc>

Sending SQL statements which update the database will return an error if QlikView has opened the ODBC connection in read-only mode.

The syntax:

```
SQL SELECT * from tabl;
```

is allowed, and will in future documentation be the preferred syntax for *Select (SQL)* (page 270), for reasons of consistency. The SQL prefix will, however, remain optional for **select** statements.

Back to *Script Statements and Keywords* (page 219).

## SQLColumns

The **sqlcolumns** statement returns a set of fields describing the columns of an *ODBC* data source, to which a **connect** has been made. These fields can be combined with the fields generated by the *SQLTables* (page 273) and *SQLTypes* (page 274) commands in order to give a good overview of a given database. The twelve 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.

The syntax is:

```
sqlcolumns
```

**Example:**

```
connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
sqlcolumns;
```

**Note!**

Some ODBC *drivers* may not support this command.  
Some ODBC drivers may produce additional fields.

Back to *Script Statements and Keywords* (page 219).

## SQLTables

The **sqltables** statement returns a set of fields describing the tables of an *ODBC* data source, to which a **connect** has been made. These fields can be combined with the fields generated by the *SQLColumns* (page 273)

and *SQLTypes* (page 274) commands in order to give a good overview of a given database. The five standard fields are:

```
TABLE_QUALIFIER
TABLE_OWNER
TABLE_NAME
TABLE_TYPE
REMARKS
```

For a detailed description of these fields, see an ODBC reference handbook.

The syntax is:

### **sqltables**

#### **Example:**

```
connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
sqltables;
```

---

#### **Note!**

Some ODBC drivers may not support this command.  
Some ODBC drivers may produce additional fields.

---

Back to *Script Statements and Keywords* (page 219).

## **SQLTypes**

The **sqltypes** statement returns a set of fields describing the types of an *ODBC* data source, to which a **connect** has been made. These fields can be combined with the fields generated by the *SQLColumns* (page 273) and *SQLTables* (page 273) commands in order to give a good overview of a given database. The fifteen 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.

The syntax is:

### **sqltypes**

#### **Example:**

```
connect to 'MS Access 7.0 Database; DBQ=C:\Course3\DataSrc\QWT.mbd';
sqltypes;
```

---

**Note!**

Some ODBC drivers may not support this command.  
Some ODBC drivers may produce additional fields.

---

Back to *Script Statements and Keywords (page 219)*.

## Star

The string specified is used to represent any values that may appear in a field, including NULL. It is used as a wildcard specifically for the QlikView application. It affects the subsequent *Load (page 253)* and *Select (SQL) (page 270)* statements. The syntax is:

```
star is [ string ]
```

where:

*string* is an arbitrary text. It can be a single symbolic character, such as an asterisk (\*). The string must be enclosed by *Quotation Marks in Scripting (page 292)* if it contains blanks

---

**Note!**

Star is is not recommended for use in your script, it is supported to maintain existing QlikView applications.

---

## Restrictions:

- The Star character is not accepted in info load files.
- The Start character cannot be used in the key fields, the fields that link tables.

## Examples of the Star statement:

```
Star is *;  
Star is %;  
Star is;
```

If nothing is specified, **star is;** is assumed; that is, there is no Star "wildcard" available. A string specified as the Star value is valid until a new **star** statement is made.

## Example of Star string used in a Load script:

```
Star is ASTERISK;  
LOAD *  
  Inline [  
    Category, Place  
    Country, United States  
    City, London  
    Country, France  
    City, Paris  
    Country, Italy  
    City, Rome  
    Country, Spain  
    City, Madrid  
    ASTERISK, Monaco  
    ASTERISK, Vatican City  
  ];
```

Back to *Script Statements and Keywords (page 219)*.

## 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 (page 253)*. The **store** statement to a CSV file does not support BIFF export.

```
store [ *fieldlist from] table into filename [ format-spec ] ;
```

*\*fieldlist*::= ( \* | *field* ) { , *field* } ) is a list of the fields to be selected. Using \* as field list indicates all fields.

*field*::= *fieldname* [**as** *aliasname* ]

*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 in Scripting (page 292)* 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.

*table* is a script labeled, already loaded table to be used as source for data.

*filename* is the name of the target file. The interpretation of file name is similar to names in **load** statements, i.e. the **directory** statements apply.

*format-spec* ::= ( ( **txt** | **qvd** ) )

The format specification consists of a the text **txt** for text files, or the text **qvd** for qvd files. If the format specification is omitted, **qvd** is assumed.

### Examples:

```
Store mytable into xyz.qvd (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;
store mytable into myfile.txt (txt);
store * from mytable into myfile.txt (txt);
```

(The two first examples have identical function.)

Back to *Script Statements and Keywords (page 219)*.

## Sub..end sub

The **sub..end 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. These can be used as any variable inside the subroutine.

*statements* is any group of one or more QlikView script statements.

Arguments are copied into the subroutine and, if the corresponding actual parameters in the **call** statement is a variable name, copied back out again upon exiting 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 and as such is ended with either a semicolon or end-of-line, each of its two clauses (**sub** and **end sub**) must not cross a line boundary.

If the sub statement is defined inside another control statement, for example a **For ... Next** loop or an **If ... Then** construction, the subroutine is only available inside the scope of that control statement. In other words: the sub statement cannot be defined inside another control statement if the subroutine call should be made from outside the control statement. This is also true if the subroutine is defined in a script file included using the **include** system variable within a control statement.

### 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)
```

The result of the above will be that locally, inside the subroutine, A will be initialized to 1, B will be initialized to 4 and C will be initialized to NULL.

When exiting the subroutine, the global variable A will get 2 as value (copied back from subroutine). The second actual parameter “(X+1)\*2” will not be copied back since it is not a variable. Finally, the global variable C will not be affected by the subroutine call.

Back to *Script Statements and Keywords (page 219)*.

## Switch..case..default..end switch

The **switch** control statement is a script selection construct forcing the script execution to follow different paths, depending on the value of an *expression*.

The syntax is:

```
switch expression { case valuelist [ statements ] } [ default statements ] end switch
```

Where:

<i>expression</i>	is an arbitrary expression.
<i>valuelist</i>	is a comma separated list of values with which the value of <i>expression</i> will be compared. Execution of the script will continue with the statements in the first group encountered with a value in <i>valuelist</i> equal to the value in <i>expression</i> . Each value in <i>valuelist</i> may be an arbitrary expression. If no match is found in any <b>case</b> clause, the statements under the <b>default</b> clause, if specified, will be executed.
<i>statements</i>	is any group of one or more QlikView script statements.

Since the **switch** statement is a control statement and as such is ended with either a semicolon or end-of-line, each of its four possible clauses (**switch**, **case**, **default** and **end switch**) must not cross a line boundary.

### 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
```

Back to *Script Statements and Keywords (page 219)*.

## Tag Field

Provides a way of assigning tags to a field. Field names not present in the document are ignored. If conflicting occurrences of a field or tag name are found, the last value is used.

A field tagged with *dimension* will be displayed at the top of all field selection controls in QlikView except in the **Edit Expression** dialog.

A field tagged with *measure* will be displayed at the top of all field selection controls in the **Edit Expression** dialog.

The keyword can be used to add tags to a field using the syntax:

```
tag fields fieldlist using mapname
```

*fieldlist* is a comma separated list of the fields that should be tagged from this point in the script.

*mapname* is the name of a mapping table previously read in a *Mapping (page 261)* load or *Mapping (page 261)* select statement.

To set individual tags, the following syntax is used:

```
tag field fieldname with tagname
```

*fieldname* is the name of the field that should be tagged.

*tagname* is the name of the tag that should be applied to the field.

### Example 1:

```
tagmap:
mapping Load * inline [
a,b
```

```
Alpha, MyTag
Num, MyTag
];
tag fields using tagmap;
```

**Example 2:**

```
tag field Alpha with 'MyTag2';
```

Back to *Script Statements and Keywords (page 219)*.

## Trace

The **trace** statement writes a *string* to the **Script Execution Progress** window and to the script log file, when used.

It is very useful for debugging purposes. Using  $\$$ -expansions of variables that are calculated prior to the Trace statement, you can customize the message.

The syntax is:

```
trace string
```

**Examples:**

```
trace Main table loaded;
```

```
Let MyMessage = NoOfRows('MainTable') & ' rows in Main Table';
```

```
trace $(MyMessage);
```

Back to *Script Statements and Keywords (page 219)*.

## Unless

The **unless** prefix and suffix is used for creating a conditional clause which determines whether a statement or exit clause should be evaluated or not. It may be seen as a compact alternative to the 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 the *exitstatement* will only be executed if *condition* is evaluated to false.

The **unless** prefix may be used on statements which already have one or several other statements, including additional **when** or **unless** prefixes.

**Examples:**

```
exit script unless A=1;
unless A=1 load * from myfile.csv;
unless A=1 when B=2 drop table Tab1;
```

Back to *Script Statements and Keywords (page 219)*.

## Unmap

The **unmap** statement disables field value *Mapping (page 261)* specified by a previous **map ... using** statement for subsequently loaded fields.

The syntax is:

```
unmap *fieldlist
```

*\*fieldlist* is a comma separated list of the fields that should no longer be mapped from this point in the script. Using \* as field list indicates all fields. The wildcard characters \* and ? are allowed in field names. Quoting of field names may be necessary when wildcards are used.

### Examples:

```
unmap Country;           disables mapping of field Country.  
unmap A, B, C;           disables mapping of fields A, B and C.  
unmap * ;                disables mapping of all fields.
```

Back to *Script Statements and Keywords (page 219)*.

## Unqualify

The **unqualify** statement is used for switching off the qualification of field names that has been previously switched on by the **qualify** statement. Refer to the documentation for the *Qualify (page 264)* statement for syntax and further information.

Back to *Script Statements and Keywords (page 219)*.

## Untag Field

Provides a way of removing tags from a field. Field names not present in the document are ignored. If conflicting occurrences of a field or tag name is found, the last value is used.

The keyword can be used to remove tags from a field using the syntax:

```
untag fields fieldlist >using mapname
```

*fieldlist* is a comma separated list of the fields which tags should be removed.

*mapname* is the name of a mapping table previously read in a *Mapping (page 261)* load or *Mapping (page 261)* select statement.

To remove individual comments, the following syntax is used:

```
untag field fieldname with tagname
```

where

*fieldname* is the name of the field that should be untagged.

*tagname* is the name of the tag that should be removed from the field.

### Example 1:

```
tagmap:  
mapping Load * inline [  
a,b  
Alpha,MyTag  
Num,MyTag
```



```
];
  untag fields using tagmap;
```

**Example 2:**

```
  untag field Alpha with MyTag2;];
```

Back to *Script Statements and Keywords (page 219)*.

## When

The **when** prefix and suffix is used for creating a conditional clause which determines whether a statement or exit clause should be executed or not. It may be seen as a compact alternative to the 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 the *exitstatement* will only be executed if *condition* is evaluated to true.

The **when** prefix may be used on statements which already have one or several other statements, including additional **when** or **unless** prefixes.

**Examples:**

```
  exit script when A=1;
  when A=1 load * from myfile.csv;
  when A=1 unless B=2 drop table Tab1;
```

Back to *Script Statements and Keywords (page 219)*.

## 24.2 Table Names

### Naming of Tables

QlikView internal 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 internal 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 or characters, should precede the first *Load* (page 253) or *Select (SQL)* (page 270) statement that creates the 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;
```

## 24.3 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. This is very useful if the same string is repeated many times in the script, e.g. a path.

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 syntax:

```
set variablename = string
```

or

```
let variable = expression
```

is used. 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.
```

The following script variables are available:

<b>Error Variables</b>	<i>page 288</i>
<b>Number Interpretation Variables</b>	<i>page 366</i>
<b>System Variables</b>	<i>page 283</i>
<b>Value Handling Variables</b>	<i>page 284</i>

**Direct Discovery System Variables**

*Direct Discovery (page 161)*

**Direct Discovery Character Variables**

*Direct Discovery (page 161)*

## System Variables

These variables have special meanings to 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.

### `$(Include =filename )`

### `$(Must_Include =filename )`

The **include** and **must\_include** variables specify a file that contains text that should be included in the script. The entire script can thus be put in a file. This is a user-defined variable.

The difference between **include** and **must\_include** is that **include** will fail silently if the file is not found during script reload, while **must\_include** will throw an error if the file is not found.

### Examples:

```
$(Include=abc.txt);
$(Must_Include=def.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.

### 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.

#### **StripComments**

If this variable is set to 0, stripping of /\*..\*/ and // comments in the script will be inhibited. Certain database drivers using /\*..\*/ comments for defining hints in **select** statements may otherwise cause script errors. It is recommended that this variable be reset to 1 immediately after the statement(s) where it is needed. If this variable is not defined, stripping of comments will always be performed.

#### **Example:**

```
set StripComments=0;
```

#### **Verbatim**

Normally all field values are automatically stripped of preceding and succeeding blank characters (ASCII 32) before being loaded into the QlikView database. Setting this variable to 1 suspends the stripping of blank characters.

#### **Example:**

```
set Verbatim = 1;
```

#### **OpenUrlTimeout**

This variable defines the timeout in seconds that QlikView should respect when getting data from URL sources (e.g. HTML pages). If omitted the timeout is about 20 minutes.

#### **Example:**

```
set OpenUrlTimeout=10
```

#### **WinPath**

Returns the browse string to Windows. A system-defined variable.

#### **WinRoot**

Returns the root directory of Windows. A system-defined variable.

## **Value Handling Variables**

#### **NullDisplay**

---

The defined symbol will substitute all NULL values from *ODBC* on the lowest level of data. A user-defined variable.

**Example:**

```
set NullDisplay='<NULL>';
```

**NullInterpret**

The defined symbol will when it occurs in a text file, Excel file or an inline statement be interpreted as NULL. A user-defined variable.

**Example:**

```
set NullInterpret=' ';
set NullInterpret =;
```

will NOT return null values for blank values in Excel (but it will for a csv text file)

```
set NullInterpret ='';
```

will return null values for blank values in Excel (but will NOT for a csv text files)

**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.

**Example:**

```
set NullValue='<NULL>';
```

**OtherSymbol**

Defines a symbol to be treated as 'all other values' before a **load/select** statement. A user-defined variable.

**Example:**

```
set OtherSymbol='+';
```

## Number Interpretation Variables

The following variables are system defined, i.e. they are automatically generated according to the current operating system settings when a new document is created. The number interpretation variables are included at the top of the script of the new QlikView document and may substitute operating system defaults for certain number formatting settings at the time of the script execution. They may be deleted, edited or duplicated freely.

**ThousandSep**

The thousands separator defined replaces the digit grouping symbol of the operating system (**Regional Settings**).

**Example:**

```
Set ThousandSep=','; (for example, seven billion must be specified as:
7,000,000,000)
```

**DecimalSep**

The decimal separator defined replaces the decimal symbol of the operating system (**Regional Settings**).

**Example:**

```
Set DecimalSep='.';
```

**MoneyThousandSep**

The thousands separator defined replaces the digit grouping symbol for currency of the operating system (**Regional Settings**).

**Example:**

```
Set MoneyThousandSep=', ';
```

**MoneyDecimalSep**

The decimal separator defined replaces the decimal symbol for currency of the operating system (**Regional Settings**).

**Example:**

```
Set MoneyDecimalSep='.';
```

**MoneyFormat**

The symbol defined replaces the currency symbol of the operating system (**Regional Settings**).

**Example:**

```
Set MoneyFormat='$ #,##0.00; ($ #,##0.00) ';
```

**DirectMoneyDecimalSep**

The decimal separator defined replaces the decimal symbol for currency in the SQL statement generated to load data using *Direct Discovery* (page 161). This character must match the character used in Direct-MoneyFormat.

Default value is '.'

**Example:**

```
Set DirectMoneyDecimalSep='.';
```

**DirectMoneyFormat**

The symbol defined replaces the currency format in the SQL statement generated to load data using *Direct Discovery* (page 161). The currency symbol for the thousands separator should not be included.

Default value is '#.0000'

**Example:**

```
Set DirectMoneyFormat='#.0000';
```

**TimeFormat**

The format defined replaces the time format of the operating system (**Regional Settings**).

**Example:**

```
Set TimeFormat='hh:mm:ss';
```

**DateFormat**

The format defined replaces the date format of the operating system (**Regional Settings**).

**Example:**

```
Set DateFormat='M/D/YY';
```

**DirectTimeFormat**

The time format defined replaces the time format in the SQL statement generated to load data using *Direct Discovery* (page 161)

**Example:**

```
Set DirectTimeFormat='hh:mm:ss';
```

**DirectDateFormat**

The date format defined replaces the date format in the SQL statement generated to load data using *Direct Discovery* (page 161)

**Example:**

```
Set DirectDateFormat='MM/DD/YYYY';
```

**TimestampFormat**

The format defined replaces the date and time formats of the operating system (**Regional Settings**).

**Example:**

```
Set TimestampFormat='M/D/YY hh:mm:ss[.fff]';
```

**DirectTimeStampFormat**

The format defined replaces the date and time format in the SQL statement generated in the SQL statement generated to load data using *Direct Discovery* (page 161)

**Example:**

```
Set DirectTimestampFormat='M/D/YY hh:mm:ss[.fff]';
```

**MonthNames**

The format defined replaces the month names convention of the operating system (**Regional Settings**).

**Example:**

```
Set MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

**LongMonthNames**

The format defined replaces the long month names convention of the operating system (**Regional Settings**).

**Example:**

```
Set LongMonthNames='January;February;March;April;May;June - -
```

**DayNames**

The format defined replaces the weekday names convention of the operating system (**Regional Settings**).

**Example:**

```
Set DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

**LongDayNames**

The format defined replaces the long weekday names convention of the operating system (**Regional Settings**).

**Example:**

```
Set Long-  
DayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
```

**ShowCalendarWeek**

This variable sets whether to show or hide calendar week numbers in AJAX and Webview. This is set to False by default.

The calendar uses ISO standard with Monday as first day of the week and week 1 as the first week with Monday to Thursday (4 days rule).

Possible values are:

'true' (or alternatively 1 or '1')

'false' (or alternatively 0 or '0')

**Examples:**

```
Set ShowCalendarWeek='true';
```

**ExponentNumberNotation**

This variable sets how to interpret values that are loaded containing exponential numbers. E and D are used as prefixes for the exponent.

Possible values are:

0 (neither 1.23E6 nor 1.23D6 is interpreted as an exponential number)

1 (1.23E6 but not 1.23D6 is interpreted as an exponential number, default value)

2 (both 1.23E6 and 1.23D6 are interpreted as an exponential number)

**Example:**

```
Set ExponentNumberNotation=2;
```

Back to *Other Functions* (page 315).

**Error Variables**

The following variables can be used for error handling in the script:

<b>ErrorMode</b>	<i>page 289</i>
<b>ScriptError</b>	<i>page 289</i>
<b>ScriptErrorDetails</b>	<i>page 290</i>



**ScriptErrorCount** *page 290*

**ScriptErrorList** *page 290*

The values of all five variables will remain after script execution. The value of the last four variables, for error handling inside the script, is of course dependent on the use of `ErrorMode=0`.

### **ErrorMode**

This variable determines what action is 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 an "Execution of script failed..." error message immediately on failure, without prompting the user for action beforehand.

#### **Example:**

```
set ErrorMode=0;
```

### **ScriptError**

Returns 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. Error codes are dual values with a numeric and a text component. The following error codes exist:

<b>0</b>	No error
<b>1</b>	General error
<b>2</b>	Syntax error
<b>3</b>	General ODBC error
<b>4</b>	General OLE DB error
<b>5</b>	General custom data- base error
<b>6</b>	General XML error
<b>7</b>	General HTML error
<b>8</b>	File not found
<b>9</b>	Database not found
<b>10</b>	Table not found
<b>11</b>	Field not found
<b>12</b>	File has wrong format
<b>13</b>	BIFF error
<b>14</b>	BIFF error encrypted
<b>15</b>	BIFF error unsupported version
<b>16</b>	Semantic error

#### **Example:**

```
set ErrorMode=0;
load * from abc.qvw;
if ScriptError=8 then
exit script;
//no file;
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 OLE DB 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.

### **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.

## 24.4 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).

---

#### **Note!**

A dollar-sign expansion is limited in how many macro expansions it can calculate. Any expansion over 1000 will not be calculated!

---

## Dollar-Sign Expansion Using a Variable

When using a variable for text replacement in the script or in an expression, the following syntax is used:

`$(variablename)`

`$(variablename)` expands to the value in *variablename*. If *variablename* does not exist the expansion will be the empty string.

For numeric variable expansions, the syntax `$(variablename)` will generate a number using the regional decimal separator, i.e. for many countries a decimal comma. Such an expansion should not be used for numbers inside the script since these must use decimal point. Instead the expansion `$(#variablename)` should be used. (Note the hash sign). It always yields a valid decimal-point representation 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.

#### **Example:**

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 stated 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 when storing scripts or parts of scripts in text files.

### Example:

```
$(include=C:\Documents\MyScript.qvs);
```

## 24.5 Quotation Marks in Scripting

You can use quotation marks in script statements in a number of different ways.

### Inside Load Statements

In a *Load* (page 253) statement, the following symbols should be used as quotation marks:

	Description	Symbol	ASCII	Example
<b>Field names</b>	double quotation marks	" "	34	"string"
	square brackets	[ ]	91, 93	[string]
	grave accents	` `	96	`string`
<b>String literals</b>	single quotation marks	' '	39	'string'

### In Select Statements

For a *Select (SQL)* (page 270) statement interpreted by the *ODBCdriver* it may be slightly different. Usually, you should use the straight double quotation marks (Alt + 0034) for field and table names, and the straight single quotation marks (Alt + 0039) for literals, and avoid using grave accents. However, some *ODBC* drivers not only accept the grave accents as quotation marks, but also prefer them. In such a case, the generated *Select (SQL)* (page 270) statements contain grave accent quotation marks.

#### Microsoft Access Quotation Marks Example

Microsoft Access *ODBC* Driver 3.4 (included in Microsoft Access 7.0) accepts the following quotation marks when analyzing the **select** statement:

**Field names and table names:** [ ] " " ` `

**String literals:** ''

Other databases may have different conventions.

### Outside Load Statements

Outside a *Load* (page 253) statement, in places where QlikView expects an expression, double quotation marks denote a variable reference and not a field reference. If you use double quotation marks, the enclosed string will be interpreted as a variable and the value of the variable will be used.

## Out-of-Context Field References and Table References

Some script functions in QlikView refer to fields that have already been created, or are in the output of a *Load* (page 253) statement, for example **Exists()** and **Peek()**. These field references are called out-of-context field references, as opposed to source field references that refer to fields that are in context, that is, in the input table of the *Load* (page 253) statement.

Out-of-context field references and table references should be regarded as literals and therefore need single quotation marks.

## Difference Between Names and Literals

The difference between names and literals becomes clearer comparing the following examples.

### Example 1:

#### 'Sweden' as Country

When this expression is used as a part of the field list in a **load** or **select** statement, the text string *"Sweden"* will be loaded as field value into the QlikView field *"Country"*.

### Example 2:

#### "land" as Country

When this expression is used as a part of the field list in a **load** or **select** statement, the content of the database field or table column named *"land"* will be loaded as field values into the QlikView field *"Country"*. This means that *land* will be treated as a field reference.

## Difference Between Numbers and String Literals

The difference between numbers and string literals becomes clearer comparing the following examples.

### Example 1:

#### '12/31/96'

When this string is used as a part of an expression, it will in a first step be interpreted as the text string *"12/31/96"*, which in turn may be interpreted as a date if the date format is 'MM/DD/YY'. In that case it will be stored in QlikView as a dual value with both a numeric and a textual representation.

### Example 2:

#### 12/31/96

When this string is used as a part of an expression, it will be interpreted numerically as 12 divided by 31 divided by 96.

See also *Script Statements and Keywords* (page 219).



## 25 Script Expressions

Expressions can be used in both the *Load* (page 253) statement and the *Select (SQL)* (page 270) 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 *ODBCdriver* and not by QlikView. However, most ODBC drivers are often capable of interpreting a number of the functions described below.

Expressions consist of *functions*, *Fields* (page 171) and *Operators* (page 787), combined in a *Script Syntax* (page 219).

All expressions in a QlikView script return a number and/or a string, whichever is appropriate. Logical functions and *Operators* (page 787) 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.

The general syntax for an expression is:

```

expression ::= (constant  constant           |
                fieldref    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 thousands separator and with a decimal point as decimal separator.

*fieldref* is a field name of the loaded table.

*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.

### 25.1 Operators

There are two types of operators in QlikView, unary operators that take only one operand, or binary operators that take two operands. Most operators are binary.

The following operators can be defined:

<b>Numeric Operators</b>	<i>page 788</i>
<b>String Operators</b>	<i>page 788</i>
<b>Logical Operators</b>	<i>page 788</i>
<b>Relational Operators</b>	<i>page 788</i>
<b>Bit Operators</b>	<i>page 789</i>

## Numeric Operators

All numeric operators use the numeric values of the operands and return a numeric value as result.

<b>+</b>	Sign for positive number (unary operator) or arithmetic addition. The binary operation returns the sum of the two operands.
<b>-</b>	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.
<b>*</b>	Arithmetic multiplication. The operation returns the product of the two operands.
<b>/</b>	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 return a string as result. The other one compares the operands and returns a boolean value to indicate match.

<b>&amp;</b>	String concatenation. The operation returns a text string, that consists of the two operand strings, one after another.
--------------	---

Example:

```
'abc' & 'xyz' returns 'abcxyz'
```

<b>like</b>	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.

<b>not</b>	Logical inverse. One of the few unary operators. The operation returns the logical inverse of the operand.
<b>and</b>	Logical and. The operation returns the logical and of the operands.
<b>or</b>	Logical or. The operation returns the logical or of the operands.
<b>Xor</b>	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 the result. All relational operators are binary.

<code>&lt;</code>	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.
<code>&lt;=</code>	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.
<code>&gt;</code>	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.
<code>&gt;=</code>	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.
<code>=</code>	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.
<code>&lt;&gt;</code>	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.
<b>precedes</b>	ASCII less than	<p>Unlike the <code>&lt;</code> operator no attempt is made to make a numeric interpretation of the argument values before the comparison. The operation returns true if the value to the left of the operator has a text representation which, in ASCII comparison, comes before the text representation of the value on the right.</p> <p><b>Example:</b></p> <pre>' 11' precedes ' 2' returns true compare this to: ' 11' &lt; ' 2' returns false</pre>
<b>follows</b>	ASCII greater than	<p>Unlike the <code>&gt;</code> operator no attempt is made to make a numeric interpretation of the argument values before the comparison. The operation returns true if the value to the left of the operator has a text representation which, in ASCII comparison, comes after the text representation of the value on the right.</p> <p><b>Example:</b></p> <pre>' 23' follows ' 111' returns true compare this to: ' 23' &gt; ' 111' returns false</pre>

## Bit Operators

All bit operators convert the operands to signed integers (32 bit) and return the result in the same way. All operations are performed bit by bit.

<b>bitnot</b>	Bit inverse.	Unary operator. The operation returns the logical inverse of the operand performed bit by bit.
<b>bitand</b>	Bit and.	The operation returns the logical AND of the operands performed bit by bit.
<b>bitor</b>	Bit or.	The operation returns the logical OR of the operands performed bit by bit.
<b>bitxor</b>	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.

## 25.2 Aggregation Functions

These functions can only be used in field lists for *Load* (page 253) statements with a **group by** clause. The aggregation functions are:

<b>Basic Aggregation Functions</b>	<i>page 298</i>
<b>String Aggregation Functions</b>	<i>page 300</i>
<b>Counter Aggregation Functions</b>	<i>page 300</i>
<b>Advanced Aggregation</b>	<i>page 301</i>
<b>Statistical Aggregation Functions in Script</b>	<i>page 302</i>
<b>Financial Aggregation Functions in Script</b>	<i>page 306</i>
<b>Statistical Test Functions in Script</b>	<i>page 307</i>

---

### Note!

If no value is found, NULL is returned for all Aggregation Functions, except Sum and Count which both return 0.

---

## Basic Aggregation Functions

### **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 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.

**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 corresponds 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 numeric value, that value is returned. Else, NULL is returned.

**Example:**

```
Load Month, only(Price) as OnlyPriceSoldFor 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, the nth value in order will be returned. 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 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 Month, MinString(Month) as FirstSalesMonth from abc.csv group by
Year;
```

### **MaxString(expression )**

Returns the last text value 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 Month, MaxString(Month) as LastSalesMonth from abc.csv group by
Year;
```

### **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. This function is only available as a script function.

#### **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. This function is only available as a script function.

#### **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;
```

For more information, see *Examples of Concat Functions (page 854)*

## 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.

### Examples:

```
Load Month, count(Sales) as NumberOfSalesPerMonth from abc.csv group
by Month;
```

```
Load Month, count(distinct Customer) as CustomerBuyingPerMonth from
abc.csv group by Month;
```

```
LoadMonth, 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.

### Example:

```
Load Month, MissingCount(Item) as NumberOfMissingItems from abc.csv
group by Month;
```

## Advanced Aggregation

There is 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 cannot 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 Analysis* (page 819) expression.

By using this function in *Add calculated dimension...* (page 622) it is possible to achieve nested chart aggregation in multiple levels. See also *Nested Aggregations and Related Issues* (page 865).

When used in chart expressions it is possible to achieve *Sum of Rows in Pivot Tables* (page 867).

### Examples:

```
aggr( sum(Sales), Country )
aggr( nodistinct sum(Sales), Country )
aggr( sum(Sales), Country, Region )
count( aggr( sum(Sales), Country ) )
```

## Statistical Aggregation Functions in Script

### **fractile(expression, fractile)**

Returns the 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;
```

### **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;
```

### **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;
```

**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;
```

**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;
```

**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 (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 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(Y,X) as Z from abc.csv group by Key;
```

**linest\_m (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 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(Y,X) as Z from abc.csv group by Key;
```

**linest\_b (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 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(Y,X) as Z from abc.csv group by Key;
```

**linest\_r2 (y-expression, x-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(Y,X) as Z from abc.csv group by Key;
```

**linest\_sem (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 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_sem(Y,X) as Z from abc.csv group by Key;
```

**linest\_seb (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 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_seb(Y,X) as Z from abc.csv group by Key;
```

**linest\_sey (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 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_sey(Y,X) as Z from abc.csv group by Key;
```

**linest\_df (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 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(Y,X) as Z from abc.csv group by Key;
```

**linest\_f (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 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(Y,X) as Z from abc.csv group by Key;
```

**linest\_ssreg (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 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(Y,X) as Z from abc.csv group by Key;
```

**linest\_ssresid (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 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(Y,X) as Z from abc.csv group by Key;
```

## Financial Aggregation Functions in Script

**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 the entire data-pair to be disregarded.

**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 the entire data-pair to be disregarded.

**Example:**

```
Load Year, npv(0.05, Payments, PayDates) as PValue from abc.csv group
by Year;
```

## Statistical Test Functions in Script

The following three functions apply to chi2-tests.

```
chi2test_p (col, row, observed_value [, expected_value])
```

returns the aggregated chi2-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 chi2-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* (*col*, *row*, *observed\_value* [, *expected\_value*]) (page 307) function 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* (*col*, *row*, *observed\_value* [, *expected\_value*]) (page 307) function 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 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 (group, value [, eq\_var = true])* (page 307) function 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 (group, value [, eq\_var = true])* (page 307) function 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 (group, value [, eq\_var = true])* (page 307) function 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 (group, value [, eq\_var = true])* (page 307) function 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 (group, value [, eq\_var = true])* (page 307) function 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 (group, value [, eq\_var = true])* (page 307) and *TTest\_conf (group, value [, sig = 0.025 [, eq\_var = true]])* (page 308) 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 (group, value [, eq\_var = true])* (page 307) and *TTest\_conf (group, value [, sig = 0.025 [, eq\_var = true]])* (page 308) 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 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* (weight, group, value [, eq\_var = true]) (page 309) function 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* (weight, group, value [, eq\_var = true]) (page 309) function 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* (weight, group, value [, eq\_var = true]) (page 309) function 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* (weight, group, value [, eq\_var = true]) (page 309) function 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* (weight, group, value [, eq\_var = true]) (page 309) function 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* (weight, group, value [, eq\_var = true]) (page 309) and *TTestlw\_conf* (weight, value [, sig = 0.025 ]) (page 312) 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* (weight, group, value [, eq\_var = true]) (page 309) and *TTestlw\_conf* (weight, value [, sig = 0.025 ]) (page 312) 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* (value) (page 310) function 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 (value)* (page 310) function 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 (value)* (page 310) function 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 (value)* (page 310) function 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 (value)* (page 310) function 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 (value)* (page 310) and *TTest1\_conf (value [, sig = 0.025 ])* (page 311) 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 (value)* (page 310) and *TTest1\_conf (value [, sig = 0.025 ])* (page 311) 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.

**TTestlw\_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, ttestlw_t(Weight, Value) as X from abc.csv group by Year;
```

**TTestlw\_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 *TTestlw\_t (weight, value)* (page 311) function for argument descriptions.

**Example:**

```
Load Year, ttestlw_df(Weight, Value) as X from abc.csv group by Year;
```

**TTestlw\_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 *TTestlw\_t (weight, value)* (page 311) function for argument descriptions.

**Example:**

```
Load Year, ttestlw_sig(Weight, Value) as X from abc.csv group by Year;
```

**TTestlw\_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 *TTestlw\_t (weight, value)* (page 311) function for argument descriptions.

**Example:**

```
Load Year, ttestlw_dif(Weight, Value) as X from abc.csv group by Year;
```

**TTestlw\_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 *TTestlw\_t (weight, value)* (page 311) function for argument descriptions.

**Example:**

```
Load Year, ttestlw_sterr(Weight, Value) as X from abc.csv group by  
Year;
```

**TTestlw\_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 *TTestlw\_t (weight, value)* (page 311) function for descriptions of the other arguments.

**Example:**

```
Load Year, ttestlw_conf(Weight, Value) as X from abc.csv group by  
Year;
```

**TTestlw\_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 *TTestlw\_t (weight, value)* (page 311) and *TTestlw\_conf (weight, value* [, *sig* = 0.025 ] (page 312) functions for argument descriptions.

**Example:**

```
Load Year, ttestlw_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 (weight, value)* (page 311) and *TTest1w\_conf (weight, value [, sig = 0.025 ])* (page 312) 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 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 (value [, sigma])* (page 313) function for argument descriptions.

returning null.

**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\_z (value [, sigma])* (page 313) function for argument descriptions.

returning null.

**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 (value [, sigma])* (page 313) function for argument descriptions.

returning null.

**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 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 (value [, sigma])* (page 313) function for descriptions of the other arguments.

returning null.

**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 the test is 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_z(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 (weight, value [, sigma])* (page 314) function 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 (weight, value [, sigma])* (page 314) function 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 (weight, value [, sigma])* (page 314) function 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 (weight, value [, sigma])* (page 314) function for descriptions of the other arguments.

**Example:**

---

```
Load Year, ztestw_conf(Weight,Value-TestValue) as X from abc.csv group
by Year;
```

*Statistical Test Functions in Script (page 307)*

## 25.3 Other Functions

In addition to the aggregation functions, QlikView offers more than 100 standard functions grouped as follows:

<b>General Numeric Functions</b>	<i>page 316</i>
<b>Script Range Functions</b>	<i>page 320</i>
<b>Exponential and Logarithmic Functions</b>	<i>page 325</i>
<b>Trigonometric and Hyperbolic Functions</b>	<i>page 325</i>
<b>Statistical Distribution Functions</b>	<i>page 326</i>
<b>Financial Functions</b>	<i>page 328</i>
<b>Mathematical Constants and Parameter-Free Functions</b>	<i>page 330</i>
<b>Counter Functions</b>	<i>page 331</i>
<b>String Functions</b>	<i>page 333</i>
<b>Mapping Functions</b>	<i>page 337</i>
<b>Inter Record Functions</b>	<i>page 338</i>
<b>Conditional Functions</b>	<i>page 340</i>
<b>Logical Functions</b>	<i>page 341</i>
<b>NULL Functions</b>	<i>page 342</i>
<b>System Functions</b>	<i>page 342</i>
<b>File Functions in Script</b>	<i>page 346</i>
<b>Table Functions</b>	<i>page 349</i>
<b>Document Functions</b>	<i>page 350</i>
<b>Date and Time Functions</b>	<i>page 350</i>
<b>Number Interpretation Variables</b>	<i>page 366</i>
<b>Formatting Functions</b>	<i>page 370</i>
<b>Color Functions</b>	<i>page 375</i>

## General Numeric Functions

In the functions below, the parameters are expressions where  $x$  should be interpreted as a real valued number.

### **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 non-negative remainder of an integer division.

#### **Examples:**

```
mod( 7,2 ) returns 1
mod( 7.5,2 ) returns NULL
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( 7.5,2 ) returns 1.5
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
```

### **numsum**(*expr1* [ , *expr2*, ... *exprN* ])

Returns the numeric sum of 1 to N arguments. As opposed to the + operator, **numsum** will treat all non-numeric values as 0.

#### **Examples:**

```
numsum( 1, 2, 4 ) returns 7
numsum( 1, 'xyz' ) returns 1
```

```
numsum( null( ) ) returns 0
```

---

**Note!**

The **numsum** function is now obsolete and is superseded by the *rangesum(expr1 [ , expr2, ... exprN ])* (page 320) function.

---

**numcount** (*expr1 [ , expr2, ... exprN ]*)

Returns the number of numeric values found in 1 to N arguments

**Examples:**

```
numcount( 1,2,4, ) returns 3
numcount( 2,xyz ) returns 1
numcount( null( ) ) returns 0
```

---

**Note!**

The **numcount** function is now obsolete and is superseded by the *rangecount(expr1 [ , expr2, ... exprN ])* (page 320) function.

---

**numavg** (*expr1 [ , expr2, ... exprN ]*)

Returns the numeric average of 1 to N arguments. If no numeric value is found, NULL is returned.

**Examples:**

```
numavg( 1,2,4 ) returns 2.33333333
numavg( 1, 'xyz' ) returns 1
numavg( null( ) 'abc' ) returns NULL
```

---

**Note!**

The **numavg** function is now obsolete and is superseded by the *rangeavg(expr1 [ , expr2, ... exprN ])* (page 320) function.

---

**nummin** (*expr1 [ , expr2, ... exprN ]*)

Returns the lowest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

**Examples:**

```
nummin( 1,2,4 ) returns 1
nummin( 1, 'xyz' ) returns 1
nummin( null( ) 'abc' ) returns NULL
```

---

**Note!**

The **nummin** function is now obsolete and is superseded by the *rangemin(expr1 [ , expr2, ... exprN ])* (page 321) function.

---

**nummax** (*expr1 [ , expr2, ... exprN ]*)

Returns the highest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

**Examples:**

```
nummax( 1,2,4 ) returns 4
nummax( 1, 'xyz' ) returns 1
nummax( null( ) 'abc' ) returns NULL
```

---

**Note!**

The **nummax** function is now obsolete and is superseded by the `rangemax(expr1 [ , expr2, ... exprN ])` (page 321) function.

**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 selected is insignificant. Non-integer items will be truncated.

**Example:**

How many combinations of 7 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 selected 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
```

**sign(x)**

Returns 1, 0 or -1 depending on whether *x* is a 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
```

**bitcount(i)**

Returns the number of set bits in *i*. Where *i* is 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( -1 ) returns 32
bitcount( n )+bitcount(bitnot n ) returns 64 (as long as n is a valid integer).
```

Back to **Other Functions**.

## Script Range Functions

Range functions will replace the following *General Numeric Functions (page 316)*: **numsum**, **numavg**, **numcount**, **nummin** and **nummax**, which should now be regarded as obsolete.

**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

```

### **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.

#### **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 found 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.

#### **Examples:**

```

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.

#### **Examples:**

```

rangekurtosis (1,2,4,7) returns -0.28571428571429

```

### **rangefractile**(*fractile*, *expr1* [ , *expr2*, ... *exprN* ])

Returns the fractile within a range of 1 to N arguments.

#### **Examples:**

```

rangefractile (0.24,1,2,4,6) returns 1

```

```
rangefractile (0.5,1,2,3,4,6) returns 3  
rangefractile (0.5,1,2,5,6) returns 3.5
```

**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 NULL 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
```

**rangeminstring**(*expr1* [ , *expr2*, ... *exprN* ])

Returns the first 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 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
```

**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'
```

**rangecorrel** (*x-value* , *y-value* { , *x-value* , *y-value* }

Returns the correlation coefficient for a series of coordinates.

*X-value* and *y-value* are single values. Each *y-value* must correspond to an *x-value*.

The function needs at least two pairs of coordinates to be calculated. Text values, null values and missing values are disregarded.

**Examples:**

```
rangecorrel (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 *Chart Inter Record Functions* (page 836) with a third optional parameter. The function needs at least one positive and one negative value to be calculated. 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 *Chart Inter Record Functions (page 836)* with a third optional parameter. 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(value { ,value} ) (page 323)* 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 *Chart Inter Record Functions (page 836)* with a third optional parameter. 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(rate, value { ,value} ) (page 324)* 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 *Chart Inter Record Functions (page 836)* with a third optional parameter. 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))
```

Back to **Other Functions**.

## Exponential and Logarithmic Functions

In the functions below, the parameters are expressions where  $x$  should be interpreted as a real valued number.

**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.

Back to **Other Functions**.

## Trigonometric and Hyperbolic Functions

In the functions below, the parameters are expressions where  $x$  should be interpreted as a real valued number. 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  $\pi$ .

**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.

Back to **Other Functions**.

## Statistical Distribution Functions

The statistical distribution functions described below are all implemented in QlikView using the Cephес library. For references and details on algorithms used, accuracy etc, please see <http://www.netlib.org/cephes/>. The Cephес function library is used by permission.

**chidist** (  $value, degrees\_freedom$  )

returns the one-tailed probability of the chi2 distribution. The Chi2 distribution is associated with a Chi2 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 chi2 distribution. *Prob* is a probability associated with the chi2 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* (  $value, degrees\_freedom$  ) (page 326) 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* (  $prob, mean, standard\_dev$  ) (page 327) function in the following way:

If  $prob = normdist(value, m, sd)$ , then  $norminv(prob, m, sd) = value$ .

### Example:

```
normdist( 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* (*value, mean, standard\_dev*) (page 326) function in the following way:

If  $\text{prob} = \text{normdist}(\text{value}, \text{m}, \text{sd})$ , then  $\text{norminv}(\text{prob}, \text{m}, \text{sd}) = \text{value}$ .

#### **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* (*prob, degrees\_freedom*) (page 327) function in the following way:

If  $\text{prob} = \text{tdist}(\text{value}, \text{df}, \text{tails})$ , then  $\text{tinvs}(\text{prob}, \text{df}) = \text{value}$ .

#### **Example:**

```
chdist( 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* (*value, degrees\_freedom, tails*) (page 327) function in the following way:

If  $\text{prob} = \text{tdist}(\text{value}, \text{df}, \text{tails})$ , then  $\text{tinvs}(\text{prob}, \text{df}) = \text{value}$ .

#### **Example:**

```
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* (*prob, degrees\_freedom1, degrees\_freedom2*) (page 327) function in the following way:

If  $\text{prob} = \text{fdist}(\text{value}, \text{df1}, \text{df2})$ , then  $\text{finvs}(\text{prob}, \text{df1}, \text{df2}) = \text{value}$ .

#### **Example:**

```
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\_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* (*value, degrees\_freedom1, degrees\_freedom2*) (page 327) function in the following way:

If  $\text{prob} = \text{fdist}(\text{value}, \text{df1}, \text{df2})$ , then  $\text{finvs}(\text{prob}, \text{df1}, \text{df2}) = \text{value}$ .

**Example:**

```
finv( 0.0019369, 8, 5 ) returns 15.0000197
```

Back to **Other Functions**.

## Financial Functions

---

**Note!**

For all financial functions it is vital that one is consistent about what units are used for specifying *rate* and *nper*. If monthly payments are made on a five-year loan at 6% annual interest, use 0.005 (6%/12) for *rate* and 60 (5\*12) for *nper*. If annual payments are made on the same loan, use 6% for *rate* and 5 for *nper*.

---

For all the arguments, cash that is paid out is represented by negative numbers. Cash received is represented by positive numbers.

Listed here are the arguments that are used in the financial functions (excepting the ones beginning with **range-**):

*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, the *pv* argument must be included.

*pv* is the present value, or 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 the *pmt* argument must be included.

*fv* is the future value, or 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 and 1 if payments are due at the beginning of the period. If *type* is omitted, it is assumed to be 0.

```
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* (page 371).

**Example:**

You are paying a new VCR by 36 monthly installments of \$20. The interest rate is 6% per annum. The bill comes at the end of every month. What is the total value of the money 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.

**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 every month. How many periods are required if the value of the money received after the last bill has been paid should equal \$786.72?

```
nper(0.005, -20, 0, 800) returns 36
```

```
pmt(rate, nper, pv [ ,fv [ , type ] ] )
```

---



Returns the payment for a loan based on periodic, constant payments and a constant interest rate. The result has a default number format of *Money* (page 371).

To find the total amount paid over the duration of the loan, multiply the returned *pmt* value by *nper*.

### Example:

The following formula returns the monthly payment on a \$20,000 loan at an annual rate of 10 percent, that must be paid off in 8 months:

```
pmt(0.1/12, 8, 20000) returns -$2,594.66
```

For the same loan, if payment is due at the beginning of the period, the payment is:

```
pmt(0.1/12, 8, 20000, 0, 1) returns -$2,573.21
```

**pv**(*rate*, *nper*, *pmt* [ , *fv* [ , *type* ] ])

Returns the present value of an investment. The result has a default number format of *Money* (page 371).

The present value is the total amount that a series of future payments is worth right now. For example, when borrowing money, the loan amount is the present value to the lender.

### Example:

What is the present value of \$100 paid to you at the end of each month during a 5 year period, given an interest rate of 7%?

```
pv(0.07/12, 12*5, 100, 0, 0) returns -$5,050.20
```

**rate**(*nper*, *pmt* , *pv* [ , *fv* [ , *type* ] ])

Returns the interest rate per period on annuity. The result has a default number format of **Fix** two decimals and %.

The *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%
```

Back to **Other Functions**.

## Black and Schole

The Black and Schole formula calculates the theoretical value of an option. In QlikView, the **BlackAndSchole** function returns the value according to the Black and Schole unmodified formula (European style options). See *The Black and Schole Formula* (page 330) below. The syntax is:

**BlackAndSchole**(*strike* , *time\_left* , *underlying\_price* , *vol* , *risk\_free\_rate* , *type*)

Where:

*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:**

BlackAndSchole(130, 4, 68.5, 0.4, 0.04, 'call') returns 11.245...

(This is 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%)

The Black and Schole Formula

**For  $x \geq 0$**

$$\int_{-\infty}^x e^{-u^2/2} du \approx \sqrt{2\pi} - (b_1 k + b_2 k^2 + b_3 k^3) e^{-x^2/2}$$

$$k = \frac{1}{1 + 0.33267x} \quad b = \begin{pmatrix} 0.4361836 \\ -0.1201676 \\ 0.937298 \end{pmatrix}$$

**For  $x < 0$**

$$f(x) = 1 - f(-x).$$

Back to **Other Functions**.

## Mathematical Constants and Parameter-Free Functions

These do not have any parameters. The parentheses are however still required.

**e ( )**

Base of the natural logarithms,  $e$ . The function returns 2.71828...

**pi ( )**

$\pi$ 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.

Back to **Other Functions**.

## Counter Functions

These functions do not have any parameters. The parentheses are however still required.

### **RecNo ( )**

Returns an integer for the number of the currently read row of the source data table. The first record is number 1.

### **RowNo ( )**

Returns an integer for the position of the current row in the resulting QlikView internal table. In contrast to **RecNo( )**, which counts the records in the raw data table, the **RowNo( )** function does not count records that are excluded by **where** clauses and is not reset when a raw data table is concatenated to another. The first row is number 1.

#### **Note!**

If you use preceding load, that is, a number of stacked **Load** statements reading from the same table, you can only use **RowNo( )** in the top **Load** statement. If you use **RowNo( )** in subsequent preceding **Load** statements, 0 is returned.

### **Examples:**

Raw data tables:

#### **Tab1.csv**

A	B
1	aa
2	cc
3	ee

#### **Tab2.csv**

A	B
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;
```

The resulting QlikView internal table:

**QVTab**

A	B	RecNo()	RowNo()
1	aa	1	1
3	ee	3	2
4	yy	2	3
6	zz	3	4

**IterNo( )**

This function is only meaningful if used together with a while clause, see *Load (page 253)*. **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.

**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 for naming each counter.

**Examples:**

```
autonumber( Region&Year&Month )
autonumber( Region&Year&Month, 'Ctrl' )
```

**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. This function is only available as a script function.

**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');
```

Back to **Other Functions**.

## String Functions

In the functions below, the parameters are expressions where *s* should be interpreted as a string.

### **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 character corresponding to number *n* in ASCII (1-127), ANSI (128-255) or Unicode (256 and above). 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*.

#### **Example:**

`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*.

#### **Example:**

`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.

#### **Example:**

`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( *textexpression* )**

Forces upper case for all data in the expression.

#### **Example:**

`upper( 'abcD' )` returns 'ABCD'.

### **lower( *textexpression* )**

Forces lower case for all data in the expression.

#### **Example:**

`lower( 'abcD' )` returns 'abcd'.

### **repeat( *s*, *n* )**

Forms a string consisting of the string *s* repeated *n* times.

#### **Example:**

`repeat( ' * ', rating )` when *rating* = 4 returns '\*\*\*\*'.

### **ltrim( *s* )**

Returns the string *s* trimmed of any leading spaces.

#### **Example:**

`ltrim( ' abc' )` returns 'abc'.

`ltrim( 'abc ' )` returns 'abc'

### **rtrim( *s* )**

Returns the string *s* trimmed of any trailing spaces.

#### **Example:**

`rtrim( ' abc' )` returns 'abc'

```
rtrim ( 'abc ' ) returns 'abc'
```

### **trim(s)**

Returns the string *s* trimmed of any leading and trailing spaces.

#### **Example:**

```
trim ( ' abc' ) returns 'abc'
trim ( 'abc ' ) returns 'abc'
trim ( ' abc ' ) returns 'abc'
```

### **subfield(s, 'delimiter' [ , index ] )**

In its three-parameter version, this script function returns a given substring from a larger string *s* with delimiter '*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 the delimiter '*delimiter*'. If several **subfield** functions are used in the same **load** statement, the Cartesian product of all combinations will be generated.

#### **Examples:**

(For 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 ''
```

### **KeepChar(s1 , s2)**

Returns the string *s1* less all characters not contained in string *s2*.

#### **Example:**

```
keepchar ( 'a1b2c3','123' ) returns 'l23'
```

### **PurgeChar(s1, s2)**

Returns the string *s1* less all characters contained in string *s2*.

#### **Example:**

```
purgechar ( 'a1b2c3','123' ) returns 'abc'
```

### **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'
```

**evaluate ( *s* )**

If the text string *s* can be evaluated as a valid QlikView expression, the evaluated result of the expression will be returned. If *s* is not a valid expression, NULL is returned.

**Note!**

This string function can not be used in chart expressions.

**Example:**

```
evaluate ( 5 * 8 ) returns 40
```

**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'
```

**Replace ( *s* , *fromstring* , *tostring* )**

Returns a string after replacing all occurrences of a given substring within the string *s* with another substring. The function is non-recursive and works from left to right.

*s* 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 the string.

**Example:**

```
replace('abccde', 'cc', 'xyz') returns 'abxyzde'
```

**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 no matching string is found, 0 is returned.

**Example:**

```
findoneof( 'my example text string', 'et%s') returns '4'  
findoneof( 'my example text string', 'et%s', 3) returns '12'  
findoneof( 'my example text string', 'x%&') returns '0'
```

**hash128 ( *expression* { , *expression* } )**

Returns a 128-bit hash of the combined input expression values. The result is a string.

**Examples:**

```
hash128 ( 'abc', 'xyz', '123' )  
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 )
```

**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 ( 'abcdefgxcdxyz', 'cd' ) returns 2
```

**applycodepage**(*text*, *codepage*)

Applies a different codepage to the field or text stated in the expression. The codepage must be in number format.

Back to **Other Functions**.

## Mapping Functions

The **ApplyMap** function is used for mapping any expression to a previously loaded mapping table. The syntax is:

```
applymap( 'mapname', expr [ , defaultexpr ] )
```

where:

*mapname* is the name of a mapping table that has previously been created through the **mapping load** or the **mapping select** statement (see *Mapping (page 261)*). Its name must be enclosed by single, straight *Quotation Marks in Scripting (page 292)*.

*expr* is the expression, the result of which should be mapped.

*defaultexpr* is an optional expression which will be used as a default mapping value if the mapping table does not contain a matching value for *expr*. If no default value is given, the value of *expr* will be returned as is.

**Examples:**

```
// Assume the following mapping table:
map1:
mapping load * inline [
x, y
1, one
2, two
3, three ] ;

ApplyMap ( 'map1', 2 ) returns 'two'
```

```
ApplyMap ('map1', 4 ) returns 4
ApplyMap ('map1', 5, 'xxx') returns 'xxx'
ApplyMap ('map1', 1, 'xxx') returns 'one'
ApplyMap ('map1', 5, null( ) ) returns NULL
ApplyMap ('map1', 3, null( ) ) returns 'three'
```

The **MapSubstring** function is used to map parts of any expression to a previously loaded mapping table. The mapping is case sensitive and non-iterative and substrings are mapped from left to right. The syntax is:

```
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 (see *Mapping (page 261)*). 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 load * inline [
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>'
```

Back to **Other Functions**.

## 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.

```
exists( field [ , expression ] )
```

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' is found in the current content 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 to the previous example.
Load Employee, ID, Salary from Employees.csv;
Load FirstName& ' ' &LastName as Employee, Comment from Citizens.csv
```

```

where exists (Employee, FirstName& ' ' &LastName);
Only comments regarding those citizens who are employees are read.
Load A, B, C, from Employees.csv where not exists (A);
This is equivalent to performing a distinct load on field A.

```

### **previous** (*expression* )

Returns the value of *expression* using data from the previous input record. In the first record of an internal table the function will return NULL. The **previous** function may be nested in order to access records further back. Data are fetched directly from the input source, making it possible to refer also to fields which have not been loaded into QlikView, i.e. even if they have not been stored in its associative 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 *fieldname* in the record specified by *row* in the internal table *tablename*. Data are fetched from the associative 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 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, see *Table Labels (page 282)*, without the ending colon. If no *tablename* is stated, the current table is assumed. If used outside the load statement or referring to another table, the table-name 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 internal table.
peek( 'Sales', -2 )
returns the value of Sales from the second last record read into the current internal 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.

```

### **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.

---

Note: This function will only work with distinct field values.

---

#### **Example:**

```

FieldValue( 'HelpText', 5 )

```

**FieldIndex**(*fieldname* , *value* )

Returns the position of the field value *value* found in the field *fieldname* (by load order). If *value* cannot be found among the field values, 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 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')
```

Back to **Other Functions**.

## 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*.

**Example:**

```
if( Amount >= 0, 'OK', 'Alarm' )
```

**alt**(*case1*[ , *case2* , *case3* , ...] , *else*)

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 field date 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*.

**Example:**

```
pick( N'A' 'B'4, , , )
    returns 'B' if N = 2
    returns 4 if N = 3
```

**match**( *str*, *expr1* [ , *expr2*,...*exprN* ] )

The **match** function performs a case sensitive comparison.

**Example:**

```
match( M, 'Jan', 'Feb', 'Mar')
    returns 2 if M = Feb
    returns 0 if M = Apr or jan
```

**mixmatch**( *str*, *expr1* [ , *expr2*,...*exprN* ] )

The **mixmatch** function performs a case insensitive comparison.

**Example:**

```
mixmatch( M, 'Jan', 'Feb', 'Mar')
    returns 1 if M = jan
```

**wildmatch**( *str*, *expr1* [ , *expr2*,...*exprN* ] )

The **wildmatch** function performs a case insensitive comparison and permits the use of wildcard characters ( \* and ? ) in the comparison strings.

**Example:**

```
wildmatch( M, 'ja*', 'fe?', 'mar')
    returns 1 if M = January
    returns 2 if M = fex
```

**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'
```

Back to **Other Functions**.

## 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).

**IsPartialReload** ( )

Returns - 1 (true) if the current reload is partial, otherwise 0 (false).

Back to **Other Functions**.

## NULL Functions

**Null** ( )

Returns a real NULL value.

**IsNull** (*expr* )

Returns -1 (true) if *expr* returns NULL, otherwise 0 (false).

Back to **Other Functions**.

## System Functions

**ClientPlatform** ( )

Returns the name of the platform the client is using, for example, *Mobile.iPhone*.

All mobile clients start with *Mobile*, a period sign and the type of mobile device. Clients using AJAX will return *Browser.Nameofbrowser*, for example *Browser.Firefox*.

**Note!**

The function will only return the client type for those clients using the QVPX protocol, that is the mobile clients and the AJAX client.

**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 entered in a **section access**.

**ComputerName** ( )

Returns a string containing the name of the computer as returned by the operating system.

**ReloadTime** ( )

Returns a timestamp for when the script last finished reexecuting.

**GetActiveSheetID** ( )

Returns a string containing the Id of the active sheet.

**GetCurrentField** (*groupname*)

returns the name of the field currently active in the group named *groupname*.

**Example:**

```
getcurrentfield( MyGroup )
```

---

```
GetCurrentSelections( [ recordsep [, tagsep [, valuesep [, max-  
values=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). *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*. Read about alternative values in *Color Codes* (page 99).

**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 )
```

**GetRegistryString(*path*, *key*)**

returns the value of a named registry *key* with a given registry *path*. This function can be used in chart and script alike.

**Example:**

```
getregistrystring('HKEY_LOCAL_MACHINE\SOFTWARE\QlikTech\QlikViewServer\Settings 7','EnableSessionLog')
```

**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.

**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;
```

**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;
```

**DocumentName ( )**

Returns a string containing the name of the current QlikView document, without path but with extension.

**DocumentPath ( )**

Returns a string containing the full path to the current QlikView document.

**DocumentTitle ( )**

Returns a string containing the title of the current QlikView document.

**GetObjectField ([*index*])**

returns the name of the dimension. *Index* is an optional integer denoting which of the used dimensions that should be returned.

**Example:**

```
getobjectfield(2)
StateName ( )
```

returns the state name of the object in which it is used. The QlikView Developer can use this function to have dynamic text and colors when an object's state is changed. It is important to note that this function works on objects only. It cannot be used in a chart expression to determine the state that the expression refers to. Below are some examples of how this function can be used.

**Example:**

```
Dynamic Text
='Region - ' & if(StateName() = '$', 'Default', StateName())
```

**Example:**

```
Dynamic Colors
if(StateName() = 'Group 1', rgb(152, 171, 206),
    if(StateName() = 'Group 2', rgb(187, 200, 179),
        rgb(210, 210, 210)
    )
)
```

**GetExtendedProperty** (*name* [, *objectid*])

returns the value of a named extended property in the sheet object with the given object id. If objectid isn't given, the sheet object containing the expression will be used.

An extended property is defined for the extension object in its definition file.

**Example:**

```
GetExtendedProperty ('Greeting')
```

Back to **Other Functions**.

## File Functions in Script

The file functions (only available in script expressions) return information about the table file which is currently being read. These functions will return NULL for all data sources except table files (Exception: **ConnectString**()).

Click on the functions that you would like to know more about:

**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 (keep the Ctrl key pressed while typing the QJPG combination), Ctrl + Q,M,P,3 for mp3 files and Ctrl + Q,W,M,A for wma files.

**Note!**

You can only read meta tags saved in the file according to the relevant specification, for example ID2v3 for MP3 files or EXIF for JPG files, not meta information saved in the Windows File Explorer.

**Example:**

```
Attribute('File', 'Title') as X,  
returns the mp3 tag 'title' in field X in each record.
```

**ConnectString**()

Returns the active **connect** string for ODBC or OLE DB connection. Returns an empty string if no **connect** statement has been executed or after a **disconnect** statement.

**filebasename**()

Returns a string containing the name of the table file currently being read, without path or extension.

**Example:**

```
Load *, filebasename() as X from  
C:\UserFiles\abc.txt  
Will return 'abc' in field X in each record read.
```

**filedir**()

Returns a string containing the path to the directory of the table file currently being read.

**Example:**

```
Load *, filedir( ) as X from
C:\UserFiles\abc.txt
Will return 'C:\UserFiles' in field X in each record read.
```

**fileextension()**

Returns a string containing the extension of the table file currently being read.

**Example:**

```
Load *, fileextension( ) as X from
C:\UserFiles\abc.txt
Will return 'txt' in field X in each record read.
```

**filename()**

Returns a string containing the name of the table file currently being read, without path but including the extension.

**Example:**

```
Load *, filename( ) as X from
C:\UserFiles\abc.txt
Will return 'abc.txt' in field X in each record read.
```

**filepath()**

Returns a string containing the full path to the table 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 read.
```

**filesize()**

Returns an integer containing the size in bytes of the file *filename* or, if no *filename* is specified, of the table 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 table 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.
```

**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('ProgramFiles')
GetFolderPath('Windows')
```

**QvdCreateTime(filename)**

Returns the XML-header time stamp from a QVD file if any (otherwise NULL).  
The *filename* is the name of a QVD file, if necessary including path.

**Examples:**

```
QvdCreateTime('MyFile.qvd')
QvdCreateTime('C:\MyDir\MyFile.qvd')
```

**QvdNoOfRecords(filename)**

Returns the number of records currently in a QVD file.  
The *filename* is the name of a QVD file, if necessary including path.

**Examples:**

```
QvdNoOfRecords ('MyFile.qvd')
QvdNoOfRecords ('C:\MyDir\MyFile.qvd')
```

**QvdNoOfFields(filename)**

Returns the number of fields in a QVD file.  
The *filename* is the name of a QVD file, if necessary including path.

**Examples:**

```
QvdNoOfFields ('MyFile.qvd')
QvdNoOfFields ('C:\MyDir\MyFile.qvd')
```

**QvdFieldName(filename, fieldno)**

Returns the name of field number *fieldno*, if it exists in a QVD file (otherwise NULL).  
The *filename* is the name of a QVD file, if necessary including path.  
The *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.  
The *filename* is the name of a QVD file, if necessary including path.

**Examples:**

```
QvdTableName ('MyFile.qvd')
QvdTableName ('C:\MyDir\MyFile.qvd')
```

Back to **Other Functions**.

**Table Functions**

The table functions return information about previously loaded data tables.

**FieldName** (*nr* , 'TableName')

Returns the name of the field with the specified number within a previously loaded table.

If the function is used within a **load** statement, it must not reference the table currently being loaded.

**Examples:**

```
LET a = FieldName(4, 'tab1');
T1:
Load a, b, c, d from abc.csv
T2:
Load FieldName (2, 'T1') Autogenerate 1;
```

**FieldNumber** ('field ' , 'TableName')

Returns the number of a specified field within a previously loaded table.

If the function is used within a **load** statement, it must not reference the table currently being loaded.

**Examples:**

```
LET a = FieldNumber('Customer', 'tab1');
T1:
Load a, b, c, d from abc.csv
T2:
Load FieldNumber ('b', 'T1') Autogenerate 1;
```

**NoOfFields** ([ 'TableName ' ])

Returns the number of fields in a previously loaded table.

**Examples:**

```
LET a = NoOfFields('tab1');
Load *, NoOfFields( ) from abc.csv;
```

**NoOfRows** ('TableName ' )

Returns the number of rows (records) in a previously loaded table.

If the function is used within a **load** statement, it must not reference the table currently being loaded.

**Examples:**

```
LET a = NoOfRows('tab1');
Load * from abc.csv where NoOfRows( )<30;
```

**NoOfTables** ()

Returns the number of tables previously loaded.

**TableName** ( ' *TableName* ' )

Returns the name of the table with the specified number.

If the function is used within a **load** statement, it must not reference the table currently being loaded.

**TableNumber** ( ' *TableName* ' )

Returns the number of the specified table.

If the function is used within a **load** statement, it must not reference the table currently being loaded.

Back to **Other Functions**.

## Document Functions

These functions can be used in charts and script.

**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.

Back to **Other Functions**.

## Date and Time Functions

In the below examples the default time and date formats hh:mm:ss and YYYY-MM-DD (ISO standard) are assumed.

**second** ( *expr* )

Second. Returns an integer representing the second when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

### Examples:

`second( '09:14:36' )` returns 36

`second( '0.5555' )` returns 55 ( Because 0.5555 = 13:19:55 )

If the time format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. To solve this problem, either change the settings or use the *Time#* (page 849) interpretation function.

**minute** ( *expr* )

Minute. Returns an integer representing the minute when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

### Examples:

`minute( '09:14:36' )` returns 14

`minute( '0.5555' )` returns 19 ( Because 0.5555 = 13:19:55 )

If the time format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. See above under *second(expr)* (page 350).

### **hour** (*expr*)

Hour. Returns an integer representing the hour when the fraction of *expr* is interpreted as a time according to the standard number interpretation.

#### **Examples:**

`hour( '09:14:36' )` returns 9

`hour( '0.5555' )` returns 13 ( Because 0.5555 = 13:19:55 )

If the time format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. See above under *second(expr)* (page 350).

### **day** (*date*)

Day. Returns an integer representing the day when the fraction of *expr* is interpreted as a date according to the standard number interpretation.

#### **Example:**

`day( '1971-10-30' )` returns 30.

If the date format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. To solve this problem, either change the settings or use the *Date#* (page 849) interpretation function.

### **week** (*date*)

Week number. Returns an integer representing the week when the fraction of *expr* is interpreted as a *date* according to the standard number interpretation.

#### **Example:**

`week( '1971-10-30' )` returns 43.

If the date format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. See above under *day(date)* (page 351).

### **month** (*date*)

Month. Returns a text string representing the month when the fraction of *expr* is interpreted as a date, but can be formatted as a number.

#### **Example:**

`month( '1971-10-30' )` returns *Oct*.

If the date format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. See above under *day(date)* (page 351).

### **year** (*date*)

Year. Returns an integer representing the year when the fraction of *expr* is interpreted as a date according to the standard number interpretation.

#### **Example:**

`year( '1971-10-30' )` returns 1971.

If the date format used does not correspond to the one set in the operating system, QlikView will not be able to make a correct interpretation. See above under *day(date)* (page 351).

### **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 **weekyear** will return different values.

**Examples:**

```
weekyear( '1996-12-30' ) returns 1997.  
weekyear( '1997-01-02' ) returns 1997.  
weekyear( '1997-12-30' ) returns 1997.  
weekyear( '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(date)* (page 351).

**weekday (date)**

Week day. Returns an integer between 0-6.

**Example:**

```
weekday( '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(date)* (page 351).

**now ( [ timer\_mode ] )**

Returns a timestamp of the current time from the system clock. The *timer\_mode* may have the following values:

- 0 Time at previously finished reload (not currently ongoing reload)
- 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 current 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' or 'UTC+hh:mm'. If no timezone is specified the local time will be returned. If *ignoreDST* is -1 (true) daylight savings time will be ignored.

**Examples:**

```
localtime ( 'Paris' )  
localtime ( 'GMT+01:00' )  
localtime ( 'Paris', -1 )  
localtime ( )
```

**MakeDate ( YYYY [ , MM [ , DD ] ] )**

Returns a date calculated from the year *YYYY*, the month *MM* and the day *DD*.

---



If no month is stated, 1(January) is assumed.

If no day is stated, 1 (the 1:st) is assumed.

### Examples:

```
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 *WW* and the day-of-week *D*.

If no day-of-week is stated, 0 (Monday) is assumed.

### Examples:

```
makeweekdate(1999,6,6) returns 1999-02-14
```

```
makeweekdate(1999,6) returns 1999-02-08
```

### **MakeTime** ( *hh* [ , *mm* [ , *ss* [ .*fff* ] ] ] )

Returns a time calculated from the hour *hh*, the minute *mm* the second *ss* with a 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 a second is stated, 000 is assumed.

### Examples:

```
maketime( 22 ) returns 22-00-00
```

```
maketime( 22, 17 ) returns 22-17-00
```

```
maketime( 22, 17, 52 ) returns 22-17-52
```

### **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'
```

### **YearToDate** ( *date* [ , *yearoffset* [ , *firstmonth* [ , *todaydate* ] ] ] )

Returns true if a *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. A negative *yearoffset* indicate previous years while a positive one 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 beginning on May 1, specify *firstmonth* = 5.

By specifying a *todaydate* (timestamp of the last script execution if omitted) it is possible to 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
```

**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* and the month replaced with *month*. *timestamp* is a standard QlikView timestamp (often just a date). *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, 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, 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, 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, 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 shift (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, 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, 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, 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, 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, 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, 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, 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 at midnight, 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
```

**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, 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, indicate a value between 2 and 12 in *first\_month\_of\_year*.

### Examples:

`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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, indicate an offset in fraction of a day in *dayoffset*, e.g 0.125 to denote 3am.

---

Note: On some processor configurations, the DayName() function can increase load time significantly. A workaround is to use the following instead:

```
TimeStamp(Floor(YourTimeStamp), 'YYYY-MM-DD')
```

---

#### **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'

### **age** (timestamp, date\_of\_birth)

Returns the age at the time of *timestamp* (in completed years) of somebody born on *date\_of\_birth*.

#### **Examples:**

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.

#### **Examples:**

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 *holiday* should be valid dates or timestamps.

#### **Examples:**

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 *holiday* should be valid dates or timestamps.

#### **Examples:**

lastworkdate ('2007-02-19', 9) returns '2007-03-01'

lastworkdate ('2006-12-18', 8, '2006-12-25', '2006-12-26') returns '2006-12-29'

### **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.

#### **Valid places and time zones:**

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.

### **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, 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, 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

Back to **Other Functions**.

## Number Interpretation Variables

The following variables are system defined, i.e. they are automatically generated according to the current operating system settings when a new document is created. The number interpretation variables are included

---

at the top of the script of the new QlikView document and may substitute operating system defaults for certain number formatting settings at the time of the script execution. They may be deleted, edited or duplicated freely.

### ThousandSep

The thousands separator defined replaces the digit grouping symbol of the operating system (**Regional Settings**).

#### Example:

```
Set ThousandSep=','; (for example, seven billion must be specified as:  
7,000,000,000)
```

### DecimalSep

The decimal separator defined replaces the decimal symbol of the operating system (**Regional Settings**).

#### Example:

```
Set DecimalSep='.';
```

### MoneyThousandSep

The thousands separator defined replaces the digit grouping symbol for currency of the operating system (**Regional Settings**).

#### Example:

```
Set MoneyThousandSep=', ';
```

### MoneyDecimalSep

The decimal separator defined replaces the decimal symbol for currency of the operating system (**Regional Settings**).

#### Example:

```
Set MoneyDecimalSep='.';
```

### MoneyFormat

The symbol defined replaces the currency symbol of the operating system (**Regional Settings**).

#### Example:

```
Set MoneyFormat='$ #,##0.00; ($ #,##0.00) ';
```

### DirectMoneyDecimalSep

The decimal separator defined replaces the decimal symbol for currency in the SQL statement generated to load data using *Direct Discovery (page 161)*. This character must match the character used in Direct-MoneyFormat.

Default value is '.'

#### Example:

```
Set DirectMoneyDecimalSep='.';
```

**DirectMoneyFormat**

The symbol defined replaces the currency format in the SQL statement generated to load data using *Direct Discovery* (page 161). The currency symbol for the thousands separator should not be included.

Default value is '#.0000'

**Example:**

```
Set DirectMoneyFormat='#.0000';
```

**TimeFormat**

The format defined replaces the time format of the operating system (**Regional Settings**).

**Example:**

```
Set TimeFormat='hh:mm:ss';
```

**DateFormat**

The format defined replaces the date format of the operating system (**Regional Settings**).

**Example:**

```
Set DateFormat='M/D/YY';
```

**DirectTimeFormat**

The time format defined replaces the time format in the SQL statement generated to load data using *Direct Discovery* (page 161)

**Example:**

```
Set DirectTimeFormat='hh:mm:ss';
```

**DirectDateFormat**

The date format defined replaces the date format in the SQL statement generated to load data using *Direct Discovery* (page 161)

**Example:**

```
Set DirectDateFormat='MM/DD/YYYY';
```

**TimestampFormat**

The format defined replaces the date and time formats of the operating system (**Regional Settings**).

**Example:**

```
Set TimestampFormat='M/D/YY hh:mm:ss[.fff]';
```

**DirectTimeStampFormat**

The format defined replaces the date and time format in the SQL statement generated in the SQL statement generated to load data using *Direct Discovery* (page 161)

**Example:**

```
Set DirectTimeStampFormat='M/D/YY hh:mm:ss[.fff]';
```



**MonthNames**

The format defined replaces the month names convention of the operating system (**Regional Settings**).

**Example:**

```
Set MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

**LongMonthNames**

The format defined replaces the long month names convention of the operating system (**Regional Settings**).

**Example:**

```
Set LongMonthNames='January;February;March;April;May;June - -
```

**DayNames**

The format defined replaces the weekday names convention of the operating system (**Regional Settings**).

**Example:**

```
Set DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

**LongDayNames**

The format defined replaces the long weekday names convention of the operating system (**Regional Settings**).

**Example:**

```
Set Long-  
DayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
```

**ShowCalendarWeek**

This variable sets whether to show or hide calendar week numbers in AJAX and Webview. This is set to False by default.

The calendar uses ISO standard with Monday as first day of the week and week 1 as the first week with Monday to Thursday (4 days rule).

Possible values are:

'true' (or alternatively 1 or '1')

'false' (or alternatively 0 or '0')

**Examples:**

```
Set ShowCalendarWeek='true';
```

**ExponentNumberNotation**

This variable sets how to interpret values that are loaded containing exponential numbers. E and D are used as prefixes for the exponent.

Possible values are:

0 (neither 1.23E6 nor 1.23D6 is interpreted as an exponential number)

1 (1.23E6 but not 1.23D6 is interpreted as an exponential number, default value)

2 (both 1.23E6 and 1.23D6 are interpreted as an exponential number)

**Example:**

---

```
Set ExponentNumberNotation=2;
```

Back to *Other Functions* (page 315).

## Formatting Functions

The formatting functions determine the display format of fields or expressions. 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 *Document Properties: Number* (page 449).

---

### Note!

For reasons of clarity all number representations are given with decimal point as decimal separator.

---

Back to **Other Functions**.

## Num

```
num(expression [ , format-code [ , decimal-sep [ , thousands-sep ] ] ] )
```

The **num** function formats the *expression* numerically according to the string given as *format-code*. Decimal separator and thousands separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the number format settings that have been specified with the *Number Interpretation Variables* (page 366) are used in first hand, in second hand the operating system number format settings are used if number interpretation variables have not been set.

### Example:

The examples below assume the two following settings:

	Default setting 1	Default setting 2
<b>Number format</b>	###0,#	#,###0.#

**num**( A, '0.0' ) where A=35648.375 returns:

	Setting 1	Setting 2
<b>String</b>	35 648 375	35648.375
<b>Number</b>	35648375	35648.375

**num**( A, '#,###0.##', '.', ',' ) where A=35648 returns:

	Setting 1	Setting 2
<b>String</b>	35,648.00	35,648.00
<b>Number</b>	35648	35648

**num**( pi(), '0,00' ) returns:

	Setting 1	Setting 2
<b>String</b>	3,14	003
<b>Number</b>	3.141592653	3.141592653

Back to **Other Functions**.

## Money

```
money(expression [ , format-code [ , decimal-sep [ , thousands-sep ] ] ] )
```

The **money** function formats the *expression* numerically according to the string given as *format-code*. Decimal separator and thousands separator can be set as third and fourth parameters. If the parameters 2-4 are omitted, the number format that is set in the operating system will be used.

### Examples:

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
<b>Money format</b>	kr # ##0,00	\$ #,##0.00

**money**( A ) where A=35648 returns:

	Setting 1	Setting 2
<b>String</b>	kr 35 648,00	\$ 35,648.00
<b>Number</b>	35648.00	35648.00

**money**( A, '#,##0 ¥', '.', ',' ) where A=3564800 returns:

	Setting 1	Setting 2
<b>String</b>	3,564,800 ¥	3,564,800 ¥
<b>Number</b>	3564800	3564800

Back to **Other Functions**.

## Date

```
date(expression [ , format-code ] )
```

The **date** function formats the *expression* as a date according to the string given as *format-code*. 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
<b>Date format</b>	YY-MM-DD	M/D/YY

**date**( A ) where A=35648 returns:

	Setting 1	Setting 2
<b>String</b>	97-08-06	8/6/97
<b>Number</b>	35648	35648

**date**( A, 'YY.MM.DD' ) where A=35648 returns:

	Setting 1	Setting 2
<b>String</b>	97-08-06	97-08-06
<b>Number</b>	35648	35648

`date( A, 'DD.MM.YY' )` where `A=35648.375` returns:

	Setting 1	Setting 2
<b>String</b>	06.08.1997	06.08.1997
<b>Number</b>	35648.375	35648.375

`date( A, 'YY.MM.DD' )` where `A=8/6/97` returns:

	Setting 1	Setting 2
<b>String</b>	NULL(nothing)	97.08.06
<b>Number</b>	NULL	35648

Back to **Other Functions**.

## Time

**time** (*expression* [ , *format-code* ])

The time function formats the expression as time according to the string given as format-code. 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
<b>Time format</b>	hh:mm:ss	hh.mm.ss

**time( A )** where `A=0.375` returns:

	Setting 1	Setting 2
<b>String</b>	09:00:00	09.00.00
<b>Number</b>	0.375	0.375

**time( A )** where `A=35648.375` returns:

	Setting 1	Setting 2
<b>String</b>	09:00:00	09.00.00
<b>Number</b>	35648.375	35648.375

**time( A, 'hh-mm' )** where `A=0.99999` returns:

	Setting 1	Setting 2
<b>String</b>	23-59	23-59

<b>Number</b>	0.99999	0.99999
---------------	---------	---------

Back to **Other Functions**.

## Dual

**dual** ( *s* , *x* )

Forced association of an arbitrary string representation *s* with a given number representation *x*. In QlikView, when several data items read into one field have different string representations but the same valid number representation, they will all share the first string representation encountered. This function can be used in scripts and chart expressions.

In scripting, 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.

### Note!

If a dual value is too large to fit in a field object, it will be represented by ## and not truncated with ... like a string.

### Example (scripting):

```
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.

Back to **Other Functions**.

## Interval

**interval** (*expression* [ , *format-code* ])

The **interval** function formats the *expression* as a time interval according to the string given as a *format-code*. 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 separator:** .

**interval**( *A* ) where *A*=0.375 returns:

<b>String</b>	09:00:00
<b>Number</b>	0.375

**interval( A )** where A=1.375 returns:

<b>String</b>	33:00:00
<b>Number</b>	1.375

**interval( A, 'D hh:mm' )** where A=1.375 returns:

<b>String</b>	1 09:00
<b>Number</b>	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:

<b>String</b>	365 09:00
<b>Number</b>	365.375

Back to **Other Functions**.

## Timestamp

**timestamp**(*expression* [ , *format-code* ])

The **timestamp** function formats the *expression* as a date and time according to the string given as *format-code*. 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
<b>Date format</b>	YY-MM-DD	M/D/YY
<b>Time format</b>	hh:mm:ss	hh:mm:ss

**timestamp( A )** where A=35648.375 returns:

	Setting 1	Setting 2
<b>String</b>	97-08-06 09:00:00	8/6/97 09:00:00
<b>Number</b>	35648.375	35648.375

**timestamp( A, 'YYYY-MM-DD hh.mm' )** where A=35648 returns:

Setting 1	Setting 2
-----------	-----------

<b>String</b>	1997-08-06 00.00	1997-08-06 00.00
<b>Number</b>	35648	35648

Back to **Other Functions**.

## Color Functions

These functions can be used in color expressions in sheet object properties supporting calculated color in image charts.

The functions **RGB**, **HSL** and **syscolor** always return a color with the alpha value of 255 (opaque).

Optionally, for any of the colors a parameter for alpha factor can be given. An alpha of 0 corresponds to full transparency. An alpha of 255 corresponds to full opacity.

### **color** ( *n* )

This function returns the color representation of color number *n* in the relevant chart palette. 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 they are defined in Visual Basic. Outside the calculated color expression in the *Chart Properties: Colors (page 652)* dialog, the function will always return black.

### **RGB** ( *e1*, *e2*, *e3* )

This function 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 they are defined in Visual Basic.

### **ARGB** ( *alpha*, *e1*, *e2*, *e3* )

This function 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 'RGB(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.

### **HSL** ( *hue*, *saturation*, *luminosity* )

This function returns the color representation of a color defined by a *hue* 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 they are defined in Visual Basic.

### **black** ( )

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** ( )

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( )**

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( )**

Returns the RGB 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( )**

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( )**

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( )**

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( )**

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( )**

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( )**

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( )**

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( )**

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( )**

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( )**

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( )**

---



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( )**

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( )**

Returns the RGB color representation for QT 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( )**

Returns the RGB color representation for QT gray (RGB 158,148,137).

**Colormix1(Value , ColorZero , ColorOne)**

This function returns an RGB color representation from a two color gradient, based on a *value* between 0 and 1.

If *value* = 0 the first color is returned.

If *value* = 1 the second color is returned.

If  $0 < \textit{value} < 1$  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])**

This function returns an RGB color representation from a two color gradient with the possibility to specify an intermediate color for the center position, based on a *value* between -1 and 1.

If *value* = -1 the first color is returned.

If *value* = 1 the second color is returned.

If  $-1 < \textit{value} < 1$  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( ) )
```

```
colormix2(x, red( ) , green( ) , black( ) )
```

The first example returns colors in a gradient from red to green, via brown. The second example returns a gradient from red to green, via black.

**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 corresponds to `COLOR_SCROLLBAR`
- 1 corresponds to `COLOR_BACKGROUND`
- 2 corresponds to `COLOR_ACTIVECAPTION`
- 3 corresponds to `COLOR_INACTIVECAPTION`
- 4 corresponds to `COLOR_MENU`
- 5 corresponds to `COLOR_WINDOW`
- 6 corresponds to `COLOR_WINDOWFRAME`
- 7 corresponds to `COLOR_MENUTEXT`
- 8 corresponds to `COLOR_WINDOWTEXT`
- 9 corresponds to `COLOR_CAPTIONTEXT`
- 10 corresponds to `COLOR_ACTIVEBORDER`
- 11 corresponds to `COLOR_INACTIVEBORDER`
- 12 corresponds to `COLOR_APPWORKSPACE`
- 13 corresponds to `COLOR_HIGHLIGHT`
- 14 corresponds to `COLOR_HIGHLIGHTTEXT`
- 15 corresponds to `COLOR_BTNFACE`
- 16 corresponds to `COLOR_BTNSHADOW`
- 17 corresponds to `COLOR_GRAYTEXT`
- 18 corresponds to `COLOR_BTNTEXT`
- 19 corresponds to `COLOR_INACTIVECAPTIONTEXT`
- 20 corresponds to `COLOR_BTNHIGHLIGHT`
- 21 corresponds to `COLOR_3DDKSHADOW`
- 22 corresponds to `COLOR_3DLIGHT`
- 23 corresponds to `COLOR_INFOTEXT`
- 24 corresponds to `COLOR_INFOBK`
- 26 corresponds to `COLOR_HOTLIGHT` (Win2000)
- 27 corresponds to `COLOR_GRADIENTACTIVECAPTION` (Win2000)
- 28 corresponds to `COLOR_GRADIENTINACTIVECAPTION` (Win2000)

Back to **Other Functions**.

## 26 Data Structures

### 26.1 Data Loading Statements

Data is loaded by **load** or **select** statements. Each of these statements generates an internal 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

The following rules apply when loading data into QlikView:

- QlikView does not make any difference between tables generated by a **load** or a **select** statement. This means that 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, which often makes it necessary to rename fields in the script. See *Renaming Fields* (page 384).

### 26.2 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 field names
5. Mapping of data if field name matches
6. Storing data in an internal table

### 26.3 System Fields

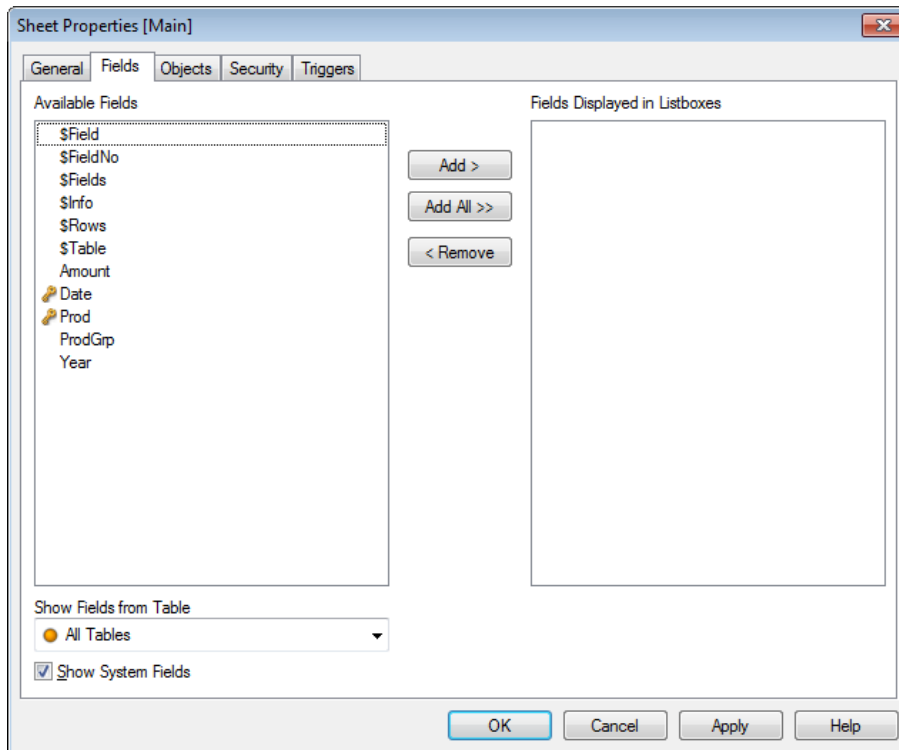
In addition to the fields extracted from the data source, system fields are also produced by QlikView. These all begin with "\$" and can be displayed in list boxes much like ordinary fields. System fields are typically created during *script* execution, are primarily used as an aid in document design.

#### Displaying System Fields

Perform the following steps:

1. Right-click on the sheet and select **System Fields**.
2. Check the **Show System Fields** box.

The system fields are now available as any other fields.



## Available System Fields

The following system fields can be displayed:

- \$Table*** Displays all internal tables loaded by the script. When a single table is selected, an information symbol will activate in the caption area of the list box. By clicking here, it is possible to view the table, if it comes from a file.
- \$Field*** Displays the fields that are read from the tables. By setting this list box to **Show Frequency** in the **List Box Properties: General** page, it is simple to detect key fields that occur in several internal tables.
- \$Fields*** The numbers in this list box represent the number of fields in different tables.
- \$FieldNo*** This list box shows the position of the fields in the tables.
- \$Rows*** This list box shows the number of rows in the tables.
- \$Info*** If info tables have been included in the document, their names will be displayed here.

## System Table

QlikView can automatically create a pivot table that makes use of the system fields. The table is called **System Table** and contains the two dimensions *\$Field* and *\$Table* and the expression *only([\$Field])*. The system table is sorted according to frequency by default. The system table object is not available for the QlikView Server clients (AJAX and Plugin clients).

## Creating a System Table

Perform the following steps:

1. Right-click on the sheet and select **New Sheet Object**.
2. Select **System Table**.

## 26.4 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:

<b>concatenate</b>	This table is concatenated with (added to) another named table or with the last previously created logical table.
<b>crosstable</b>	This table is converted from crosstable format to column format.
<b>generic</b>	This table is split into several other logical tables.
<b>info</b>	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.
<b>intervalmatch</b>	The table (which must contain exactly two columns) is interpreted as numeric intervals, which are associated with discrete numbers in a specified field.
<b>join</b>	This table is joined by QlikView with another named table or with the last previously created logical table, over the fields in common.
<b>keep</b>	This table is reduced to the fields in common with another named table or with the last previously created logical table.
<b>mapping</b>	This table (which must contain exactly two columns) is read as a mapping table, which is never associated with other tables.
<b>semantic</b>	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 *Table Viewer (page 183)*.

## 26.5 Associations between Logical Tables

### Data Associations

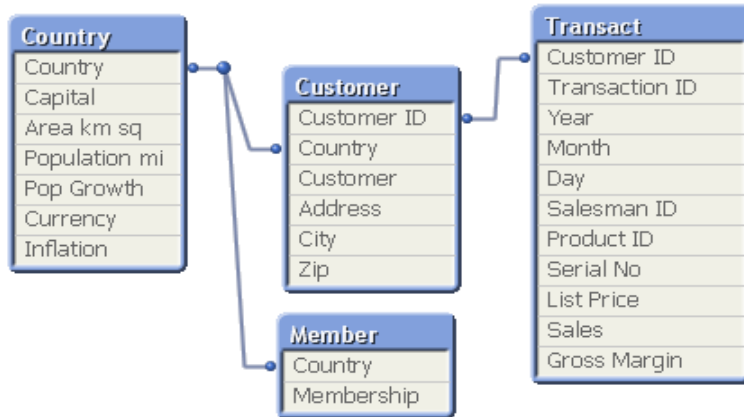
A database can have many tables. Each table can be considered as a list of something, i.e. each record in the list represents an instance of an object of some type.

#### Example:

If two tables are lists of different things, for example if one is a list of customers and the other a list of invoices, and the two tables have a field such as the customer number in common, 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, for example selecting a field value in a list box.

This means that a QlikView association is almost the same thing as a QlikView join. 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.



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.

## QlikView Association Compared to SQL Natural Outer Join

A QlikView association resembles 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.

### Limitations for Associating Fields

1. It is not possible to 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 activated. 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.

### Workaround

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 the charts.

## Synthetic Keys

When two or more internal 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.

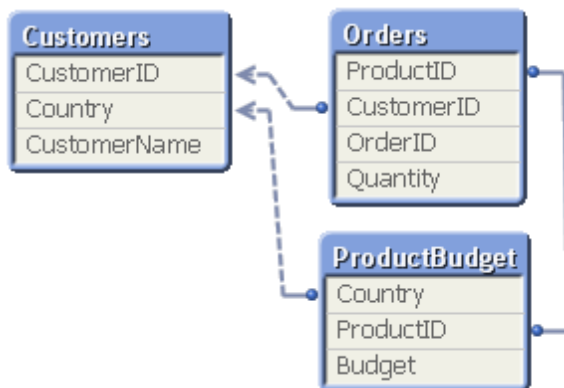
An overall analysis of the intended table structure by the application designer is recommended, including the following:

- Forming the own non-composite keys, typically using string concatenation inside an **AutoNumber** script function.
- Making sure only the necessary fields connect. If for example a date is used as a key, make sure not to load e.g. *year*, *month* or *day\_of\_month* from more than one internal table.

## Circular References

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.



*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.

## Adjusting Automatically Loosely Coupled Tables

If you prefer that another table is loosely coupled that QlikView's default choice, you can modify it in two different ways:

- Declare a specific table as loosely coupled via a **loosen table** statement in the script. See section *Loosen Table (page 260)* for more information about this statement.
- Select **Document Properties: Tables** after the script has been executed and change the setting for loosely coupled tables. See chapter *Circular References (page 111)* for more information about circular references and loosely coupled tables.

## 26.6 Renaming Fields

Renaming fields in the script is something that all QlikView developers do, not only for creating links between tables, but also for making the sometimes cryptic database field names understandable for the users.

### Reasons to Rename Fields

The following examples explain some of the situations when you would need to rename fields.

#### Example 1:

Two fields are named differently although they denote the same thing:

- The field *ID* in the *Customers* table
- The field *CustomerID* in the *Orders* table

The two fields denote a specific customer identification code and should both be named the same, for example *CustomerID*.

#### Example 2:

Two fields are named the same but actually denote different things:

- The field *Date* in the *Invoices* table
- The field *Date* in the *Orders* table

The two fields should preferably be renamed, to for example *InvoiceDate* and *OrderDate*.

#### Example 3:

There may be errors such as 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 corrections are made.

### Renaming Fields Using Different Methods

Since fields can be renamed in the script, there is no need to change the original data. There are different methods to rename fields.

#### Using the As Specifier

The most common way is to create aliases inside the *Load* (page 253) or *Select (SQL)* (page 270) statements, by using the **as** specifier.

#### Example:

```
Load ID as CustomerID, Name, Address, Zip, City, State from Customer.csv;
```

An advantage of this method is that it makes the script easy to understand for other developers. If there are many fields to rename, you may not want to use this method, because of the work needed to enter all new field names manually in the script.

**Tip!** You can rename fields directly in the preview pane when you are loading data using the **File Wizard**. This automatically adds the **as** specifier to the script. See *File Wizard: Type* (page 198).

#### Using the Alias or Qualify Statement

The *Load* (page 253) or *Select (SQL)* (page 270) statement can be preceded by an *Alias* (page 223) or *Qualify* (page 264) statement. These two statements are similar, in that they “silently” rename the fields in the output of the **Load** or **Select** statement.

#### Example:

```
Alias ID as CustomerID;
```



```
Load * from Customer.csv;
```

However, a consequence of using the **Alias** or **Qualify** statement is that you cannot use a resident load that refers to an original field name – instead it must refer to the field name as defined in the **Alias** or **Qualify** statement. This is sometimes confusing – especially if the **Alias** or **Qualify** statement is written earlier in the script, far from the **Load** statement. The **Alias** or **Qualify** statement will make your script harder to understand for other developers.

## Using the Rename Fields Statement

This is a very good method if you want to rename all or some fields at the end of the script run. The best way is to use a mapping table with the old and new field names and use this as follows.

### Example:

```
FieldNameMap;
Mapping Load OldFieldName, NewFieldName From FieldNames ;
Rename Fields using FieldNameMap;
```

You can store the mapping table in your database or in an Excel sheet so that it is easier to maintain.

A good solution may be to use a combination of using the **as** specifier and using the **Rename fields** statement. The **as** qualifier is used to define the data model, and the **Rename fields** statement to make the fields user-friendly. See *Rename Field (page 265)* and *Mapping (page 261)* for more information.

## Renaming Sheet Objects

You can change the labels of list boxes and other sheet objects while keeping the logical associations defined by the field names intact. Select **Properties** from the object menu of the sheet object to rename it.

## 26.7 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 internal 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

Even 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 table.

### Example:

```
load a, b, c from table1.csv;
concatenate load a, c from table2.csv;
```

The resulting internal 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 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 **nonconcatenate** 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
```

## 26.8 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 internal table. In some situations this is needed, but there are disadvantages:

- 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 explicit joins needs to be used.

---

**Note!**

In this manual the term join is usually used for joins made before the internal tables are created. The association, made after the internal 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 internal table with another named table or with the last previously created 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 internal 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 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 explicit joins need to be used.

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	ee	—

```

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.

## 27 Evaluating the Loaded Data

### 27.1 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.

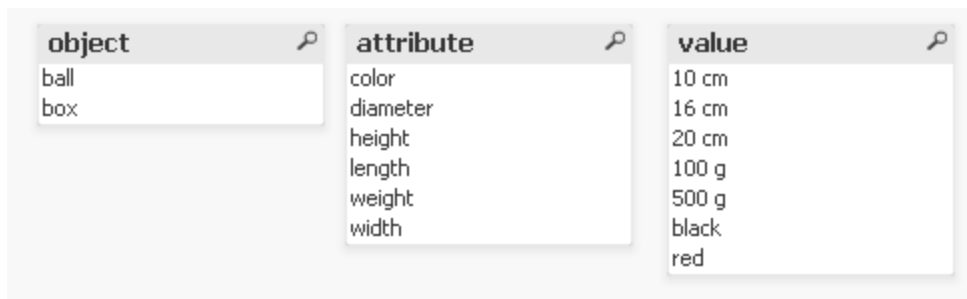
Consider the example below. It is a generic database containing two objects, a ball and a box. Obviously some of the attributes, like color and weight, are common to both the objects, while others, like diameter, height length and width are not.

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

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.

If this database is loaded into QlikView using the standard way, we 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.



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.

## 27.2 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.xlsx

Year	Jan	Feb	Mar	Apr	May	Jun
2008	45	65	78	12	78	22
2009	11	23	22	22	45	85
2010	65	56	22	79	12	56
2011	45	24	32	78	55	15
2012	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 **crosstable** prefix to the **load** or **select** statement.

The statement for loading this cross table could be:

```
crosstable (Month, Sales) load * from ex1.xlsx ;
```

The result in QlikView would be as follows:



Year	Month	Sales
2008	Jan	11
2009	Feb	12
2010	Mar	15
2011	Apr	22
2012	May	23
	Jun	24
		32
		35
		45
		55
		56
		65
		68
		78
		79
		82
		85

The cross table is often preceded by a number of qualifying columns, which should be read in a straight-forward way. This is the case in example 2:

### Example 2:

#### ex2.xlsx

Salesman	Year	Jan	Feb	Mar	Apr	May	Jun
A	2008	45	65	78	12	78	22
A	2009	11	23	22	22	45	85
A	2010	65	56	22	79	12	56
A	2011	45	24	32	78	55	15
A	2012	45	56	35	78	68	82
B	2008	57	77	90	24	90	34
B	2009	23	35	34	34	57	97
B	2010	77	68	34	91	24	68
B	2011	57	36	44	90	67	27
B	2012	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.xlsx ;
```

The result in QlikView would be:

Salesman	Year	Month	Sales
A	2008	Apr	11
B	2009	Feb	12
	2010	Jan	15
	2011	Jun	22
	2012	Mar	23
		May	24
			27
			32
			34
			35
			36
			44
			45
			47
			55
			56
			57
			65

For a description of the syntax see *Crosstable* (page 229).

## 27.3 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?

#### Table OrderLog

Start	End	Order
01:00	03:35	A
02:30	07:58	B
03:04	10:27	C
07:23	11:43	D

#### Table 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
0:00	0	Start of shift 1	-	-	-
1:18	1	Line stop	A	1:00	3:35
2:23	2	Line restart 50%	A	1:00	3:35
4:15	3	Line speed 100%	B	2:30	7:58
4:15	3	Line speed 100%	C	3:04	10:27
8:00	4	Start of shift 2	C	3:04	10:27
8:00	4	Start of shift 2	D	7:23	11:43
11:43	5	End of production	D	7:23	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. Non-numeric limits render the interval to be disregarded (undefined) while NULL limits extend the interval indefinitely (unlimited).
- The intervals may be overlapping and the discrete values will be linked to all matching intervals.

## 27.4 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 res-
ident 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	2011-01-21	Southwest
000120	2013-01-06	Southwest
000120	2013-03-05	Northwest
000120	2013-03-05	Southwest
000110	-	Northwest
000120	-	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:

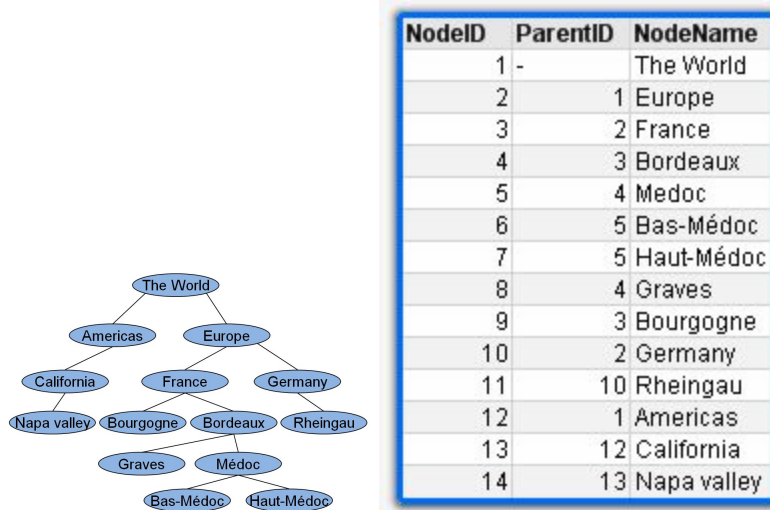
Key	Name	Date	Sales
000110	Spengler Aaron	2009-08-18	100
000110	Spengler Aaron	2009-12-25	200
000110	Spengler Aaron	2011-02-03	300
000110	Spengler Aaron	2011-05-05	400
000120	Ballard John	2011-06-04	500
000120	Ballard John	2013-01-20	600
000120	Ballard John	2013-03-10	700
000120	Ballard John	2013-03-13	800
000120	Ballard John	2013-09-21	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	-	2011-01-20	2009-08-18	100
000110	Northwest	Spengler Aaron	-	2011-01-20	2009-12-25	200
000110	Southwest	Spengler Aaron	2011-01-21	-	2011-02-03	300
000110	Southwest	Spengler Aaron	2011-01-21	-	2011-05-05	400
000120	Northwest	Ballard John	-	2013-01-05	2011-06-04	500
000120	Southwest	Ballard John	2013-01-06	2013-03-04	2013-01-20	600
000120	Northwest	Ballard John	2013-03-05	-	2013-03-10	700
000120	Northwest	Ballard John	2013-03-05	-	2013-03-13	800
000120	Northwest	Ballard John	2013-03-05	-	2013-09-21	900

## 27.5 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.



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. It is recommended that you have a single top node that links the entire hierarchy, such as The World in this example, for performance and structural reasons.

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 a 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 (page 242)* for more information.

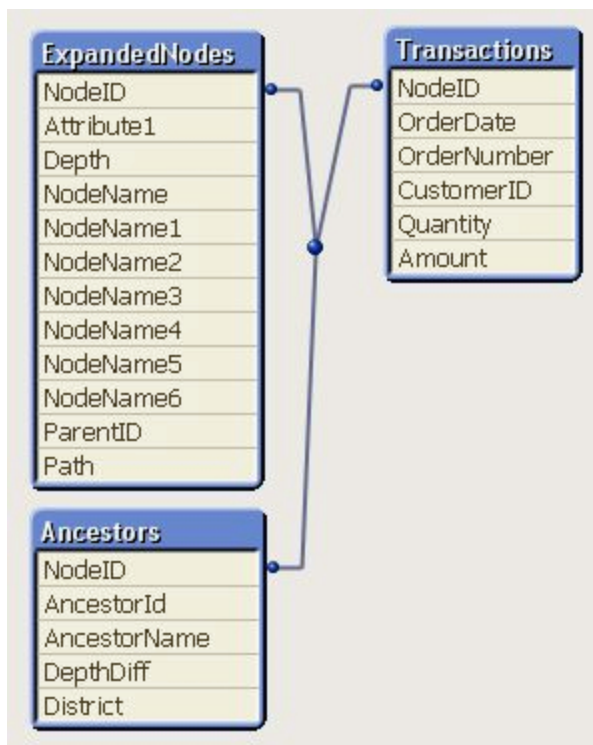
NodeID	ParentID	NodeName	Level0	Level1	Level2	Level3	Level4	Level5
1	-	The World	The World	-	-	-	-	-
2	1	Europe	The World	Europe	-	-	-	-
3	2	France	The World	Europe	France	-	-	-
4	3	Bordeaux	The World	Europe	France	Bordeaux	-	-
5	4	Medoc	The World	Europe	France	Bordeaux	Medoc	-
6	5	Bas-Médoc	The World	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 it is not easy to use the level fields for searches or selections, since a prior knowledge is needed about which level to 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 *Hierarchy Parameters (page 215)* 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.



## 27.6 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 clicking 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.

Relation
Predecessor
Successor

In the *presidents* example in the QlikView examples directory, the script to generate the links *Predecessor* and *Successor* could be:

```
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

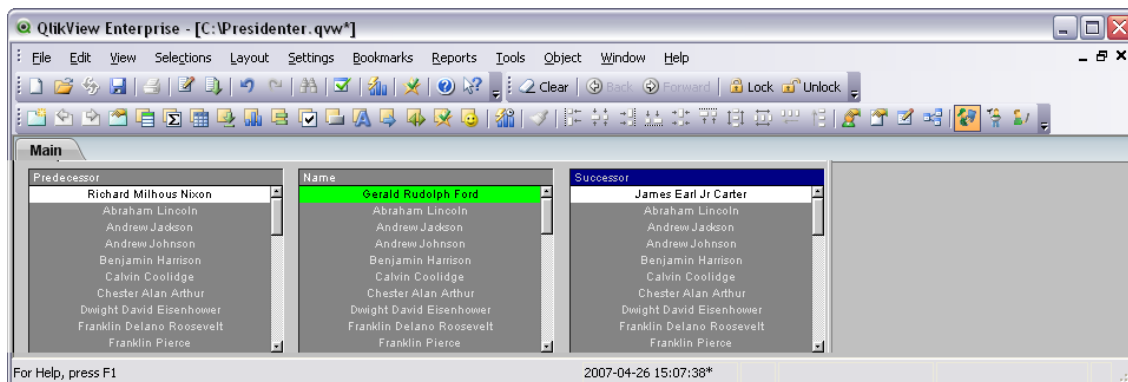
Note also that this **load** statement contains two fields labeled No and two fields labeled Relation. Such a **load** statement would cause a script execution error if used to load an internal table since the load procedure for one single internal table demands that none of the fields have the same 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, the following script is needed:

```
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*.

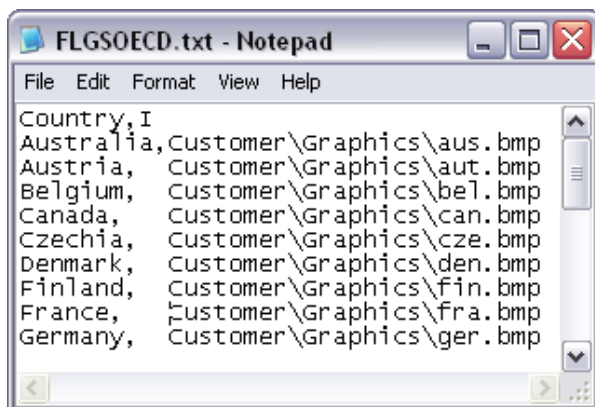
Two different **semantic** statements are needed since we want two different list boxes with relations.

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.

## 27.7 Linking Information to Field Values

Information in the form of text files, images or external application files can be associated to data in a Qlik-View document. To use this feature tables has to be created, 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 below.



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 *OtherSymbol* (page 404)), however, is allowed.

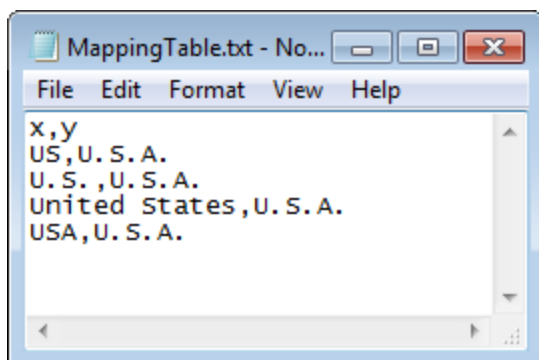
Using **bundle info load**, it is possible to bundle external data into a QlikView document. Read more about *Bundle* (page 224).

## 27.8 Data Cleansing

When loading data from different tables, 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, the problem needs to be solved. 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.



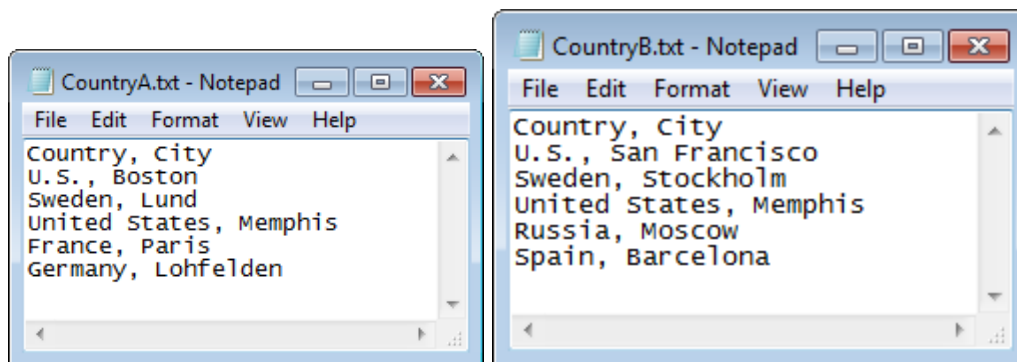
*Example of a mapping table*

## 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 internal 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.



To avoid the occurrence of three different records denoting the United States in the concatenated table, create a table similar to that shown 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 *Mapping (page 261)*) loads the file *MappingTable.txt* as a mapping table with the label *CountryMap*.

The **map** statement (for a detailed description of the syntax see *Map ... using (page 261)*) 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 *Concatenating Several Tables into One (page 385)*), 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 Qlik-View 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 *Unmap (page 280)*).

For mapping on expression level, use the **applymap** function (see *Mapping Functions (page 337)*).

For mapping on substring level, use the **mapsubstring** function (see *Mapping Functions (page 337)*).

## 27.9 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*, *NTNAME* 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 *Star (page 275)*.

### OtherSymbol

In many cases a way to represent all other values in a table is needed, 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 internal 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:

##### Table Customers

###### CustomerID Name

1	ABC Inc.
2	XYZ Inc.
3	ACME INC
+	Undefined

##### Table Orders

---

**CustomerID Name**

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!!

## 27.10 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 **nullasvalue** statement. See *NullAsValue* (page 262). 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 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
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)

## 27.11 Character Set

Character set is a file specifier for the **Load** statement that defines the character set used in the file.

Available character sets are:

- ansi**
- oem**
- mac**
- utf8**
- unicode**

**codepage is N**

The file can be written with the **ansi** character set (Windows), with the **oem** character set (DOS, OS/2, AS400 and others), **unicode**, **utf8** or with the **mac**. The conversion from the **oem** character set is not implemented for MacOS. With the codepage specifier, it is possible to use any Windows codepage. If nothing is specified, codepage 1252 is assumed under Windows.

The **ansi**, **oem** and **mac** specifiers were used in older versions of QlikView and will still work. However, they will not be generated when creating the **Load** statement with a modern QlikView.

**Examples:**

```
LOAD * FROM a.txt (utf8, txt, delimiter is ',', embedded labels)
```

```
LOAD * FROM a.txt (unicode, txt, delimiter is ',', embedded labels)
```

```
LOAD * FROM a.txt (codepage is 10000, txt, delimiter is ',', no  
labels)
```

Back to *Script Statements and Keywords (page 219)*.



---

## 28 QVD Files

A QVD (QlikView Data) file is a file containing a table of data exported from QlikView. QVD is a native QlikView format and can only be written to and read by QlikView. The file format is optimized for speed when reading data from a QlikView script but it is still very compact. Reading data from a QVD file is typically 10-100 times faster than reading from other data sources.

QVD files can be read in two modes: standard (fast) and optimized (faster). The selected mode is determined automatically by the QlikView script engine. Optimized mode can be utilized only when all loaded fields are read without any transformations (formulas acting upon the fields), though the renaming of fields is allowed. A WHERE clause causing QlikView to unpack the records will also disable the optimized load.

A QVD file holds exactly one data table and 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.

### 28.1 Purpose of QVD Files

QVD files can be used for many purposes. At least four major uses can be easily identified. More than one may apply in any given situation:

#### Increasing Load Speed

By buffering non-changing or slowly changing blocks of input data in QVD files, script execution becomes considerably faster for large data sets.

#### Decreasing Load on Database Servers

The amount of data fetched from external data sources can also be greatly reduced. This reduces work load on external databases and network traffic. Furthermore, when several QlikView scripts share the same data it is only necessary to load it once from the source database into a QVD file. The other applications can make use of the same data via this QVD file.

#### Consolidating Data from Multiple QlikView Applications

With the *Binary* (page 223) script statement it is possible to load data from only one single QlikView application into another one, but 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 for facilitating incremental load, i.e. exclusively loading new records from a growing database.

To learn more about the use of QVD files and Incremental Load, follow the link below:

*Using QVD Files for Incremental Load* (page 410)

### 28.2 Create QVD Files

A QVD file can be created by one of three different methods:

1. Explicit creation and naming using the *Store (page 276)* command in the QlikView script. Simply state in the script that a previously read table or part thereof is to be exported to an explicitly named file at a location of your choice.
2. Automatic creation and maintenance from script. By preceding a **load** or **select** statement with the *Buffer (page 223)* prefix, QlikView will automatically create a QVD file which under certain conditions can be used instead of the original data source when reloading data.
3. Explicit, manual creation and naming from layout or via *Internal Macro Interpreter (page 893)*. Data can be 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.

There is no difference between the resulting QVD files, e.g. with regard to reading speed etc.

## 28.3 Reading Data from QVD Files

A QVD file can be read into or accessed by QlikView by the following methods:

1. Loading a QVD file as an explicit data source. QVD files can be referenced by a **load** statement in the QlikView script just like any other type of text files (csv, fix, dif, biff etc). The *File Wizard: Type (page 198)* handles QVD files by the same principles.

### Examples:

```
load * from xyz.qvd (qvd);
```

```
load Name, RegNo from xyz.qvd (qvd);
```

```
load Name as a, RegNo as b from xyz.qvd (qvd);
```

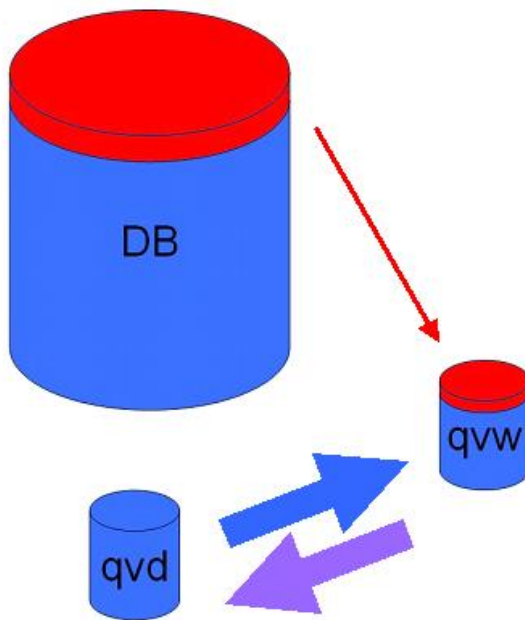
2. Automatic loading of buffered QVD files. When using the *Buffer (page 223)* prefix on **load** or **select** statements, no explicit statements for reading are necessary. QlikView will determine the extent to which it will use data from the QVD file as opposed to acquiring data via the original **load** or **select** statement.
3. Accessing QVD files via the script. A number of script functions (all beginning with **qvd**) can be used for retrieving various information on the data found in the XML header of a QVD file. These functions are presented under the *File Functions in Script (page 346)* functions.

## 28.4 Using QVD Files for Incremental Load

Incremental load is a very common task in relation to data bases. It is defined as loading nothing but new or changed records from the database. All other data should already be available, in one way or another. With *QVD Files (page 409)* it is possible to perform incremental load in most cases.

The basic process is described below:

1. Load the **new data** from Database table (a slow process, but loading a limited number of records).
2. Load the **old data** from QVD file (loading many records, but a much faster process).
3. Create a new QVD file.
4. Repeat the procedure for every table loaded.



The complexity of the actual solution depends on the nature of the source database, but the following basic cases can be identified:

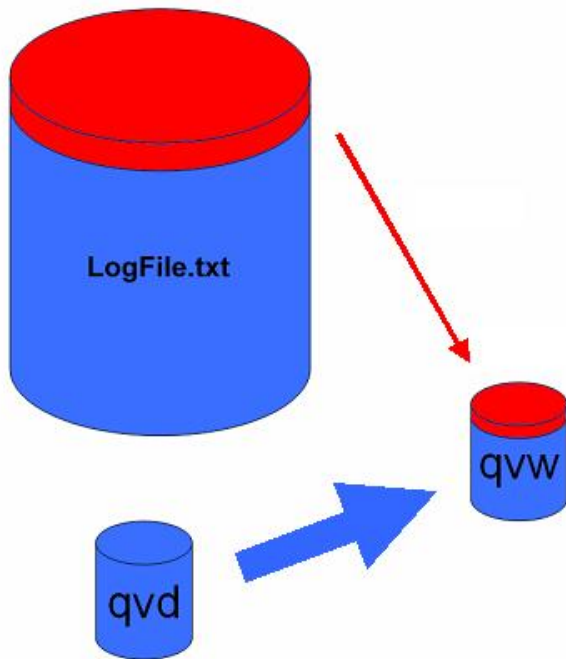
- 1) *Case 1: Append Only (page 411)* (typically log files)
- 2) *Case 2: Insert Only (No Update or Delete) (page 412)*
- 3) *Case 3: Insert and Update (No Delete) (page 413)*
- 4) *Case 4: Insert, Update and Delete (page 414)*

Below you will find outlined solutions for each of these cases. The reading of QVD files can be done in either optimized mode or standard mode. (The method employed is automatically selected by the QlikView script engine depending on the complexity of the operation.) Optimized mode is (very approximately) about 10x faster than standard mode or about 100x faster than loading the database in the ordinary fashion.

## Case 1: Append Only

The simplest case is the one of log files; files in which records are only appended and never deleted. The following conditions apply:

- The database must be a log file (or some other file in which records are appended and not inserted or deleted) which is contained in a text file (no ODBC/OLE DB).
- QlikView keeps track of the number of records that have been previously read and loads only records added at the end of the file.

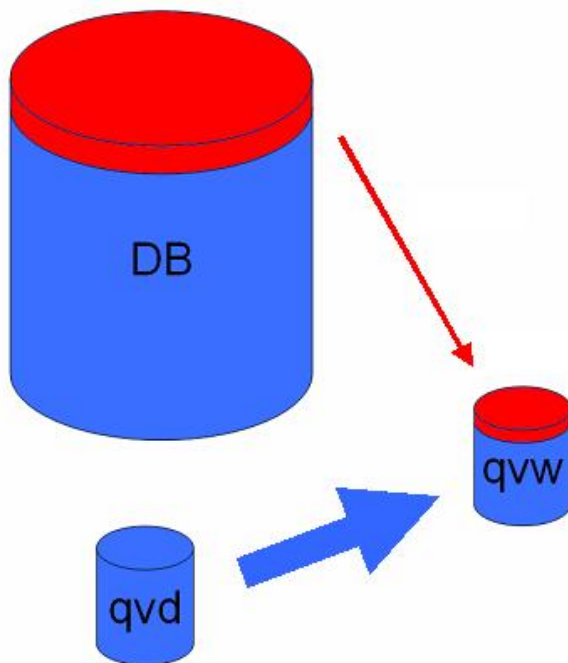
**Script Example:**

```
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. However, the problem can still be solved with minimum amount of extra work. The following conditions apply:

- 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 recognize 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;

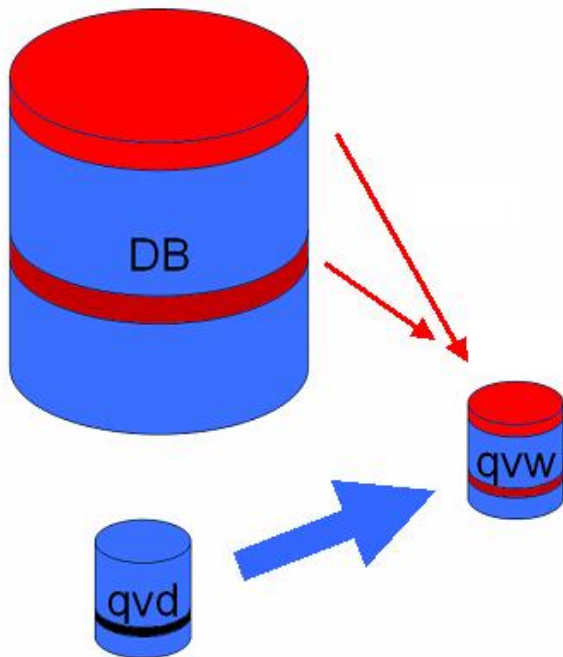
```

(The hash signs in the SQL WHERE clause define the beginning and end of a date. Check your database manual for the correct date syntax for your database.)

### Case 3: Insert and Update (No Delete)

The next case is applicable when data in previously loaded records may have changed between script executions. The following conditions apply:

- 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 recognize which records are new.
- A primary key field is required for QlikView to sort out updated records from the QVD file.
- This solution will force the reading of the QVD file to standard mode (rather than optimized), which is still considerably faster than loading the entire database.



### 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);

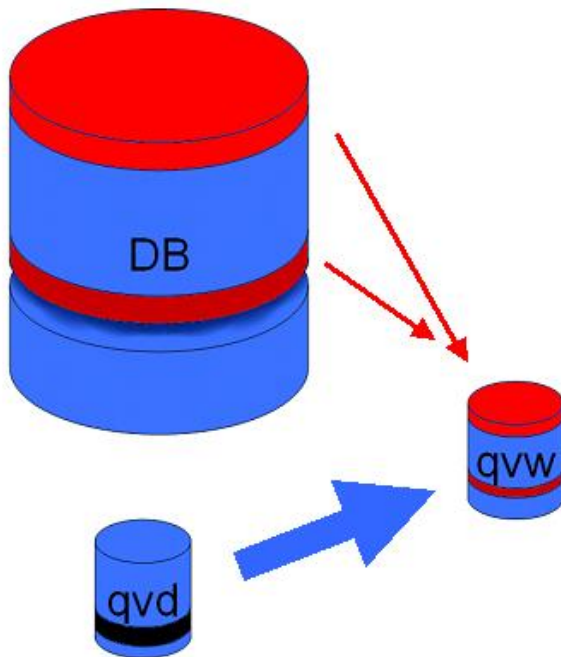
STORE QV_Table INTO File.QVD;

```

### 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 following conditions apply:

- 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 recognize which records are new.
- A primary key field is required for QlikView to sort out updated records from the QVD file.
- This solution will force the reading of the QVD file to standard mode (rather than optimized), which is still considerably faster than loading the entire database.

**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(PrimaryKey);
```

```
Inner Join SQL SELECT PrimaryKey FROM DB_TABLE;
```

```
If ScriptErrorCount = 0 then  
STORE QV_Table INTO File.QVD;  
Let LastExecTime = ThisExecTime;  
End If
```





## 29 Security

A security mechanism in QlikView 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.

### 29.1 Authentication and Authorization

Authentication is any process by which it is verified 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.

### 29.2 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.

### 29.3 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.

### 29.4 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**.

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... ..
```

## 29.5 Access Levels in Section Access

Access to QlikView documents can be authorized for specified users or groups of users. In the security table, users can be assigned to the access levels ADMIN or USER. If no access level is assigned, the user cannot open the QlikView document.

A person with ADMIN access can change everything in the document. Using the **Security** page in the **Document Properties** and **Sheet Properties** 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.

---

**Note!**

ADMIN rights are only relevant for local documents! Documents opened on a Server are always accessed with USER rights.

---

## 29.6 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.

None, all, or any combination of the security fields may be loaded in the access section. It is thus not necessary to use USERID – an authorization can be made using other fields, e.g. serial number only.

<b>ACCESS</b>	A field that defines what access the corresponding user should have.
<b>USERID</b>	A field that should contain an accepted user ID. QlikView will prompt for a User ID and compare it to the value in this field. This user ID is not the same as the Windows user ID.
<b>PASSWORD</b>	A field that should contain an accepted password. QlikView will prompt for a Password and compare it to the value in this field. This password is not the same as the Windows password.
<b>SERIAL</b>	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.

---

<b>NTNAME</b>	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.
<b>NTDOMAINSID</b>	A field that should contain a string corresponding to a Windows NT Domain SID. Example: S-1-5-21-125976590-4672381061092489882 QlikView will fetch the logon information from the OS and compare it to the value in this field.
<b>NTSID</b>	A field that should contain a Windows NT SID. Example: S-15-21-125976590-467238106-1092489882-1378 QlikView will fetch the logon information from the OS and compare it to the value in this field.
<b>OMIT</b>	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.

---

**Note!**  
You should not apply OMIT on key fields, as this will change the underlying data structure. This may create logical islands and calculation inconsistencies.

---

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 should be converted to upper case using the **upper** function, see *upper(textexpression)* (page 334), 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.

A wildcard, i. e. \*, 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*, *NTNAME* 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.

---

**Note!**

When loading data from a QVD file, the use of the upper function will slow down the loading speed.

---

**Note!**

To generate access tables in inline statements use the *Access Restriction Table Wizard* (page 195).

---

**Note!**

If you have enabled section access, you cannot use the section access system field names listed here as field names in your data model.

---

**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”.

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

## 29.7 Restrictions on QlikView Functionality

The controls found on the *Document Properties: Security (page 438)* page and the *Sheet Properties: Security (page 467)* page make 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.

## 29.8 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 **Document Properties: Opening** page 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 only those records connected to REDUCTION=3.

## 29.9 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.

### 29.10 Encryption

The communication between a QlikView Server and a QlikView Windows client is encrypted. If, however, 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.



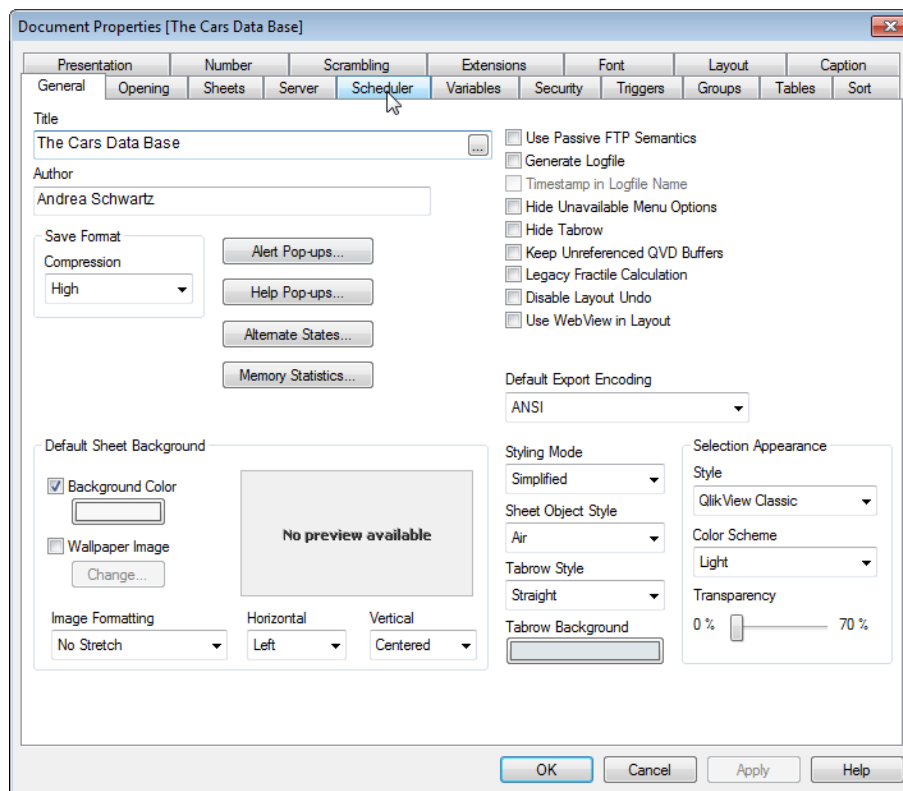
---

# Part 5 Sheet and Sheet Objects

## 30 Document Properties

Opens the *Document Properties: General* (page 424) page. Here the document specific parameters, such as opening picture and sound, default font, sort order etc. can be specified for this specific document.

### 30.1 Document Properties: General



*Document Properties, General*

**Title** Here the title that is to be displayed in the window caption can be entered. As default the document file name is used. To alter it, enter the selection in the **Title** edit box.

**Author** In the **Author** edit box specify the author of the QlikView file.

**Save Format** In this group set the level of compression for the QlikView files.

#### **Compression**

The options **Medium** and **High** substantially decrease the size of the QlikView file while sacrificing some load and save speed. Compression options set under **Document Properties** affect the current document. Compression options set under **User Preferences** affect all new documents. For further details on compression, and how to set default compression for all new files, see *User Preferences: Save* (page 73) in the **User Preferences** dialog.

**Alert Pop-ups** This button opens the **Pop-up Window Settings** dialog where the default look of alert pop-up messages can be determined.



---

<b>Help Pop-ups</b>	This button opens the <i>Pop-Up Window Settings (page 427)</i> dialog where the default look of help pop-up messages can be determined.
<b>Alternate States...</b>	Alternate States functionality is enabled by a QlikView developer. The developer invokes the <b>Alternate States</b> dialog by clicking this button. For more information see <i>Alternate States (page 113)</i> .
<hr/>	
<b>Note!</b>	
Do not enable this functionality unless it is really needed in the QlikView Document.	
<hr/>	
<b>Memory Statistics</b>	Click this button to save a table 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.
<b>Default Sheet Background</b>	The <b>Background Color</b> for the <b>Default Sheet Background</b> of the document window(s) can be customized by checking this alternative. A color can be defined as a solid color or a gradient via the <i>Color Area (page 428)</i> dialog that opens when clicking the button. Alternatively, it is possible to specify a background image by checking <b>Wallpaper Image</b> and clicking the <b>Change</b> button. This will open the <b>Change Wallpaper</b> dialog where a picture file can be selected. The background image can be altered with the following options.
<b>Image Formatting:</b>	
<b>No Stretch:</b>	
The image is displayed as is, which may cause masking problems.	
<b>Fill:</b>	
The image is stretched in order to fill the sheet, without regard to aspect ratio.	
<b>Keep Aspect:</b>	
The image is stretched as far as possible while retaining the correct aspect ratio.	
<b>Fill with Aspect:</b>	
The image is stretched as far as possible while retaining the correct aspect ratio. Areas not covered are then filled by cropping of the image.	
<b>Tile:</b>	
If this option is selected, the image will be tiled the as many times as space allows.	
 <b>Horizontal and Vertical orientation:</b>	
<b>Horizontal:</b>	
The image can be horizontally aligned: <b>Left</b> , <b>Centered</b> or <b>Right</b> .	
<b>Vertical:</b>	
The image can be vertically aligned: <b>Top</b> , <b>Centered</b> or <b>Bottom</b> .	
<b>Use Passive FTP Semantics</b>	If the <b>Use Passive FTP Semantics</b> option is checked, the passive Ftp semantics will be enabled. Ftp comes into play in the <i>Open Internet Files or Open QlikView Document (page 193)</i> . The passive semantics option is used for communicating with a server through a firewall.
<b>Generate Logfile</b>	To track the execution of the load script select <b>Generate Logfile</b> . The information that is shown in the <b>Progress</b> window during script execution will be stored as qv.log.

---

<b>Timestamp in Logfile Name</b>	Puts a timestamp in the logfile name, e.g. <i>sales.qvw.2009_02_26_12_09_50.log</i> . The setting is only available if the <b>Generate Logfile</b> check-box is marked.
<b>Hide Unavailable Menu Options</b>	The setting <b>Hide Unavailable Menu Options</b> is used in conjunction with the security settings. Any menu options that should normally be grayed out, will instead be completely hidden from the user.
<b>Hide Tabrow</b>	The QlikView sheet tabrow may be removed by checking <b>Hide Tabrow</b> . In order to move between sheets use the <i>Actions (page 552)</i> (Function: <b>Next Sheet / Previous Sheet</b> ) or the <b>Sheet Toolbar</b> of the <i>View Menu (page 52)</i> . It is also possible to use automation control, which will e.g. make it possible to force the user to move between sheets and even making selections in a specific order.
<b>Keep Unreferenced QVD Buffers</b>	The <b>Keep Unreferenced QVD Buffers</b> setting overrules the normal procedure whereby any automatically created <i>QVD Files (page 409)</i> are purged immediately after script execution, provided they are no longer used by the document that created them. This setting should normally be left unchecked.
<b>Legacy Fractile Calculation</b>	If the <b>Legacy Fractile Calculation</b> check box is enabled, QlikView will use discrete values as results from the fractile aggregation function. Deselect the check box and QlikView will use an interpolated value, as known from the Percentile function in Microsoft Excel. QlikView versions before 7.5 have only supported the discrete value algorithm.
<b>Disable Layout Undo</b>	When the check box <b>Disable Layout Undo</b> 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.
<b>Use WebView in Layout</b>	Toggles <b>WebView</b> mode, which uses the internal web browser in QlikView to display the document layout as an AJAX page, on and off.
<b>Default Export Encoding</b>	Use <b>Default Export Encoding</b> to set the default character set for export in new documents. Choose one of these option: ANSI, Unicode or UTF-8.
<b>Styling Mode</b>	In the drop-down menu <b>Styling Mode</b> choose the object style for all the sheet objects.  The <b>Advanced</b> mode makes it possible to configure several settings, such as the <b>Sheet Object Style</b> and the type of border to be used, set on the <b>Layout</b> page of the objects.  The <b>Simplified</b> mode also gives the choice of <b>Sheet Object Style</b> , but configures most settings automatically. It sets such things as <b>Scrollbar Style</b> and borders. Some settings can still be changed on the <i>Layout (page 492)</i> page of the objects.
<b>Sheet Object Style</b>	In the drop-down <b>Sheet Object Style</b> select a style for sheet object captions. The style selected will be used for all sheet objects with a caption in the document.

**Tabrow Style**

Under **Tabrow Style** select one of the available styles for the tab row appearance in this drop-down. The style selected will be used for all tabs in the document.

**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 **Led Checkboxes** styles all use color coding for indication of selected, possible and excluded values. The **Windows Checkboxes** style and the **LED Checkboxes** style mimic the standard Windows interface with a check box at each value. By choosing a specific style in this control, it is possible to force the document to appear in one style or another wherever opened. More information about the selection styles can be found under *User Preferences (page 69)* and *List Box Properties: Presentation (page 487)*. When using the selection styles based on color, there are a number of different color schemes available. The basic color scheme (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 from among 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** on the computer where it is opened.

**Color Scheme**

Sets the selection color scheme for the document. Select from among 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** on the computer where it is opened.

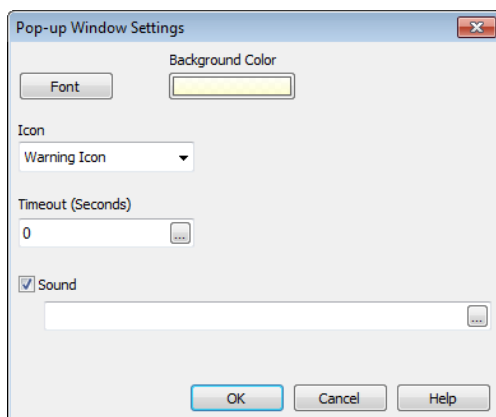
**Transparency**

Sets the transparency of the selection color in list boxes and multi boxes.

**Tabrow Background**

Specify a custom color to the background of the tab row by clicking the corresponding **Tabrow Background** button.

## Pop-Up Window Settings



*The Pop-up Window Settings dialog*

This dialog can be accessed from either the **Document Properties: General** or the **Alerts** dialog pages, both under the **Settings** menu. It is used for setting the properties of either **Help Pop-ups** or **Alert Pop-ups**.

Through the controls of the dialog the following properties can be customized:

The **Font** button opens the standard *Font (page 491)* dialog, allowing the font settings for the pop-up text to be changed.

The **Background Color** of the pop-up window can be defined as a solid color or a gradient via the *Color Area (page 428)* dialog that opens when clicking the colored button.

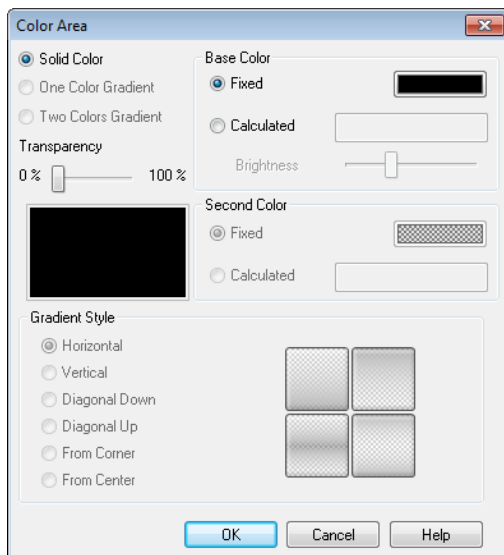
The default **Icon** for the pop-up is a **Question Icon** for **Help Pop-ups** and a **Warning Icon** for **Alert Pop-ups**, respectively. Choose between several other options or just set the display to **No Icon** from the spinner box.

Enter a time delay in the **Timeout** edit box to have pop-up windows automatically close after a set time delay (This delay is measured in milliseconds. By setting the timeout to 0 the pop-up will stay on the screen until the user clicks it away).

Check **Sound** in order to link a sound to the pop-up, see *Linking Information to Field Values (page 401)*.

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 QlikView sound (*Internal Files (page 157)*) already bundled with the program (e.g. qmem://<bundled>/sounds/qv\_ok.wav).

## Color Area



### *The Color Area dialog*

Most colored surfaces in the QlikView layout can be formatted from the **Color Area** dialog. Specify either a solid color covering the entire area or a color gradient effect. The color gradient options described below are unavailable for some purposes and will then be grayed out in the dialog, e.g. for text colors.

- Solid Color**                      The basic option using a single uniform **Base Color**.
- One Color Gradient**            This option introduces the use of varying **Brightness** to create a color gradient effect.
- Two Colors Gradient**            A color gradient effect is created between the **Base Color** and a specified **Second Color**.

Color(s) can be fixed or calculated as specified in the **Base Color** and **Second Color** groups.

Any color settings can be previewed in the dialog.

In the **Base Color** group the base color are determined for solid surfaces and gradients.

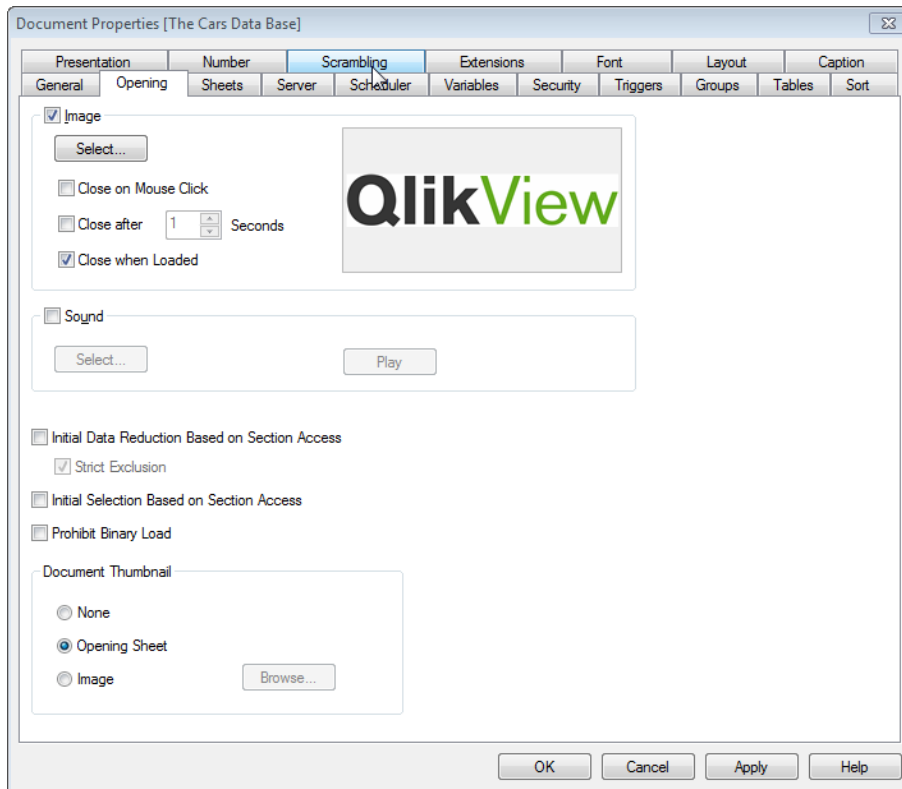
- |                   |  |
|-------------------|--|
| <b>Fixed</b>      | This alternative specifies a single, basic color. Click the colored button to change the color.  |
| <b>Calculated</b> | The color may be dynamically calculated from an expression. The expression must be a valid color representation, which is achieved by using the <i>Color Functions</i> (page 375). Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas. If the result of the expression is not a valid color representation, the program will default to black. |
| <b>Brightness</b> | When using a <b>One Color Gradient</b> this slider determines the relative shading/brightness of the color at the opposite end of the gradient. The slider setting goes from darker (left) to brighter (right). At the center position the setting translates as a uniform color.  |

In the **Second Color** group the second color is determined for two-color gradients.

- |                   |  |
|-------------------|--|
| <b>Fixed</b>      | This alternative specifies a single, basic color. Click the colored button to change the color.          |
| <b>Calculated</b> | The color may be dynamically calculated from an expression, using the <i>Color Functions</i> (page 375). |

If **One Color Gradient** or **Two Colors Gradient** was selected, the direction of the gradient effect is determined by settings in the **Gradient Style** group. It is possible to achieve different results by combining the available alternatives (**Horizontal**, **Vertical** etc.) with clicking one of the four button areas, that represent gradient direction.

## 30.2 Document Properties: Opening



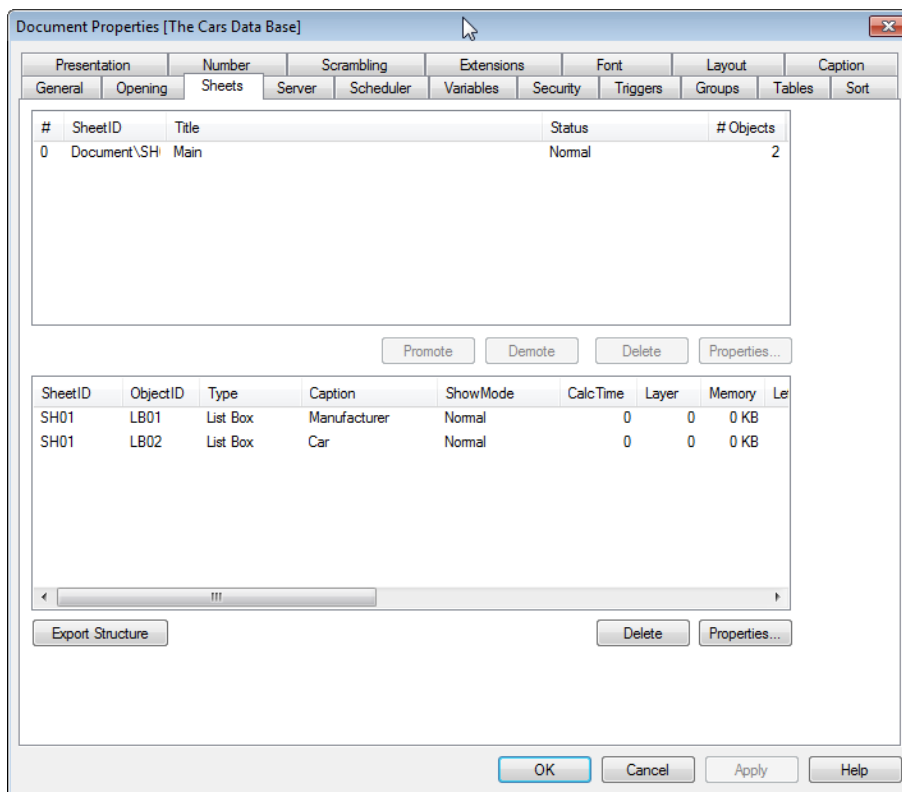
### *Document Properties, Opening*

On this tab it is possible to customize the opening of a document.

- |                              |   |
|------------------------------|---|
| <b>Image</b>                 | Mark this check box to have an opening image for the document.  |
| <b>Select...</b>             | Click this button to open the <b>Select Image</b> dialog where it is possible to select an image (bmp, jpg, jpeg or png). |
| <b>Close on Mouse Click</b>  | Lets the user close the opening image by clicking it. Either this check box or the one below must be marked.              |
| <b>Close after N Seconds</b> | Closes the opening image automatically. Either this check box or the one above must be marked.                            |
| <b>Close when Loaded</b>     | Closes the opening image when the file has been loaded.   |
| <b>Sound</b>                 | Mark this check box to add an opening sound for the document. This makes the <b>Select</b> button available.              |
| <b>Select...</b>             | This button opens the <b>Select Sound File</b> dialog where it is possible browse for a sound file.                       |
| <b>Play</b>                  | Click this button to play the sound that is selected.   |

<b>Initial Data Reduction Based on Section Access</b>	<p>If the <i>Security (page 417)</i> 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.</p> <p>When this feature is used on a document not run exclusively on a QlikView Server it should always be combined with the <b>Prohibit Binary Load</b> option (see below) in order to maintain data access security. This setting is mutually exclusive with <b>Initial Selection Based on Section Access</b>.</p>
<b>Strict Exclusion</b>	<p>If this check box is marked in combination with <b>Initial Data Reduction Based on Section Access</b>, 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 enabled by default in documents created with QlikView 7.02 and later and is recommended for maximum access security.</p>
<b>Initial Selection Based on Section Access</b>	<p>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. When this feature is used on a document not run exclusively on a QlikView Server it should always be combined with the <b>Prohibit Binary Load</b> option (see below) in order to maintain data access security. This setting is mutually exclusive with <b>Initial Data Reduction Based on Section Access</b>.</p>
<b>Prohibit Binary Load</b>	<p>If this option is selected it will not be possible to load data from the document's qvw file via a <i>Binary (page 223)</i> statement in another QlikView document.</p>
<b>Document Thumbnail</b>	<p>Sets how the document will be displayed on the AccessPoint when the option <b>Thumbnail</b> is chosen.</p> <p><b>None</b> A thumbnail with the message <b>Missing Image</b> will be displayed on the AccessPoint.</p> <p><b>Opening Sheet</b> The opening sheet of the document will be displayed on the AccessPoint.</p> <p><b>Image</b> Click on the <b>Browse</b> button to find the image to display on the AccessPoint.</p>

## 30.3 Document Properties: Sheets



### *Document Properties, Sheets*

The **Document Properties: Sheets** tab helps to keep track of every sheet and sheet object of the QlikView-document. This dialog page consists of two lists, the **Sheets** list and the **Sheet Objects** list.

It is possible to sort either table on any column, by clicking its column header.

The **Sheets** list contains the following information:

- #** The sheet number (position), counting from 0.
- SheetID** The unique **ID** of the sheet.
- Title** The text currently shown in the sheet tab.
- Status** Indicates whether the sheet is normal, hidden or has a show condition.
- # Objects** The current number of sheet objects on the sheet.

After clicking a sheet in the list, use the buttons below to either **Delete** it or open its **Properties** dialog directly from this menu. It is also possible to **Promote/Demote** the sheet from here.

The **Sheet Objects** list contains the following information:

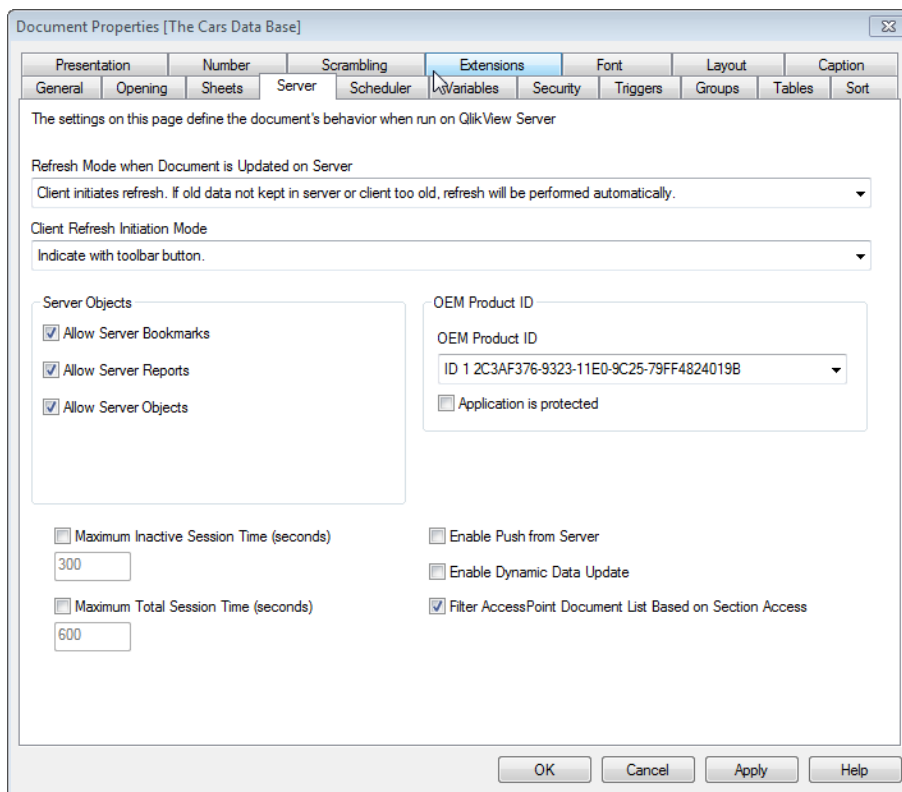
- SheetID** The unique **ID** of the sheet.
- ObjectID** Lists the unique **Object ID** of every existing sheet object. Linked objects have the same ID.



<b>Type</b>	Clarifies the object type.
<b>Caption</b>	The text currently shown in the caption of the sheet object.
<b>ShowMode</b>	Indicates whether the object is normal, minimized, maximized, hidden or has a show condition.
<b>CalcTime</b>	The time in milliseconds needed for the last recalculation of the object's contents. This information is only relevant for objects that are actually calculated (charts, Table boxes and any objects containing formulas).
<b>Layer</b>	The layer as defined on the <b>Layout</b> page of a sheet object properties dialog, defined as <b>Bottom</b> , <b>Normal</b> or <b>Top</b> corresponding to the numbers -1, 0 or 1 in the list. (The layer of a sheet object can however be set to any number between -128 and +127 via the Automation API.)
<b>Memory</b>	The amount of transient memory (in kBytes) needed for the last recalculation of the object's contents. This information is only relevant for objects that are actually calculated (charts, Table boxes and any objects containing formulas).
<b>Left, Top, Right, Bottom</b>	The current position of the indicated edge of the sheet object relative to the edge of the document window, expressed in pixels.
<b>Width, Height</b>	The size dimensions of the object, expressed in pixels.
<b>Left (Min), Top (Min), Right (Min), Bottom (Min)</b>	The current position of the object when iconized (=minimized), expressed in pixels.
<b>Width</b>	The current width of the sheet object's minimized icon, expressed in pixels.
<b>Height</b>	The current height of the sheet object's minimized icon, expressed in pixels.
<b>Export Structure</b>	By pressing this button it is possible to export the sheet and sheet object structure of the document to a set of text files. These text files, one for the sheets ( <i>filename.Sheets.tab</i> ), and one for the sheet objects ( <i>filename.Objects.tab</i> ) can easily be read back into QlikView for further analysis with the full power of the QlikView logic. A dialog will appear where the target folder for the export can be chosen. The default is to put the files in the same folder as the QlikView document.

After clicking an object in the list, use the buttons below to either **Delete** it or open its **Properties** dialog directly from this menu. Multiple objects can be selected with Ctrl-click.

## 30.4 Document Properties: Server



### *Document Properties, Server*

On this tab 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.**

Choose this option if the server should never 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 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 session will simply be terminated.

**Client initiates refresh. If old data not kept in server or client too old, refresh will be performed automatically**

Choose this option if it is preferred 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** (see above).

- |   |  |
|---|--|
| <b>Indicate with toolbar button</b>                 | The availability of new data is indicated by the <b>Refresh</b> button in the toolbar turning green and being enabled (and the corresponding <b>File</b> menu command being enabled). The refresh is performed at a time of the user's choice by clicking on the toolbar button or by means of the <b>File</b> menu <b>Refresh</b> command.  |
| <b>Prompt with dialog</b>                           | In addition to the <b>Refresh</b> 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 on the toolbar button or by means of the <b>File</b> menu <b>Refresh</b> command. |
| <b>Just do it but tell the user it is happening</b> | The refresh will be initiated as soon as the server indicates that new data is available. A dialog will be shown to indicate that the refresh is taking place.   |
| <b>Just do it</b>                                   | The refresh will be initiated as soon as the server indicates that new data is 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.   |

### Server Objects

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.

- |                               |   |
|-------------------------------|---|
| <b>Allow Server Bookmarks</b> | 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. Additionally, <b>Allow Server Objects</b> must be enabled. |
| <b>Allow Server Objects</b>   | 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.  |
| <b>Allow Server Reports</b>   | 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.  |

### OEM Product ID

The settings in this group make it possible to add an OEM Product ID to the document. The OEM Product ID helps ensure that only OEM customers can open the document. This option is only available with an OEM Partner License.

- |                       |   |
|-----------------------|---|
| <b>OEM Product ID</b> | Select <b>OEM Product ID</b> from the drop-down list. |
|-----------------------|---|

### QlikView Server Timeouts

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 allows this to be set 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.

<b>Maximum Inactive Session Time (seconds)</b>	This check box enables an inactive session timeout, i.e. the Server will terminate a session where the user has not clicked in a given time. The length of the inactive session timeout is set in seconds.
<b>Maximum Total Session Time (seconds)</b>	This check box enables a total session timeout, i.e. the Server will terminate a session after a given time, regardless of whether the user has clicked or not. The length of the total session timeout is set in seconds.
<b>Enable Push from Server</b>	Mark this check box if the Server should allow graceful document refresh.
<b>Enable Dynamic Data Update</b>	Mark this check box if the Server should allow dynamic updates in a document.
<b>Filter AccessPoint Document List Based on Section Access</b>	When this option is enabled, users will only see those documents in the document list on the AccessPoint and in Open in Server, he or she has access to based on section access, in addition to other access rights. Note that those names present in the NTNAME column in section access will be saved as clear text in the qvw file, however, no passwords will be saved.  Note: Even when this option is enabled, the AccessPoint may in some cases list documents that the user does not have access to based on Section Access. This is because of how the Section Access functionality works. However, the user can only see such documents, not open them.

## 30.5 Input Box Properties: Constraints and Document Properties: Variables

*Input Box Properties, Constraints and Document Properties, Variables*

The **Input Box Properties: Constraints** page is opened by right-clicking on an input box and choosing the **Properties** command in the float menu. The *Variable* (page 173) that have been added to **Displayed Variables** in the *Input Box Properties: General* (page 541) page are listed in the **Variables** list on the **Constraints** tab where it can be modified.

The **Document Properties: Variables** page is accessed from **Settings - Document Properties: Variables**. The list of variables may be modified by the **Show System Variables** option. The **New...** button adds a new variable to the document. **Delete** button removes the selected variable.

In the **Settings for Selected Variable** group the edit box displays the current **Value** of the selected variable. The value may be entered as a calculated formula. Click on the **...** button to open the *Edit Expression Dialog* (page 757).

Variables are normally not included in bookmarks, but here it is possible to check the option **Include in Bookmarks**.

In the **Input Constraints** group 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 may be shown. The following alternatives are available:

<b>No Constraints</b>	Input values are not checked against any specified constraints.
<b>Standard</b>	Input values will be checked against one of a number of common standard constraints selected from the drop-down box. As default no constraint at all is selected, i.e. any value can be entered into the variable. Only one of the alternatives: <b>Standard</b> , <b>Custom</b> , <b>Predefined Values Only</b> and <b>Read-only</b> can be checked.
<b>Custom</b>	Input values will be checked against a user specified constraint. The constraint is entered in the edit box and must 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.
<b>Predefined Values Only</b>	Input values will be checked against a list of values defined in the <b>Predefined Values</b> group. Only input values found in the list will be accepted.
<b>Read-only</b>	Marks the variable as read-only. No values can be entered.
<b>Enable Edit Expression Dialog</b>	Check this alternative to enable editing of the variable value in the <b>Edit Expression</b> dialog, which is opened from the <b>...</b> button that appears when clicking on the value itself.
<b>Sound on Illegal Entry</b>	Check this alternative to have QlikView issue a sound warning when the user attempts to enter a value which falls outside of the constraints.
<b>Error Message</b>	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 checking this alternative it is possible to specify a <i>Custom Error Messages</i> (page 875) 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 <i>Calculated Formula</i> (page 871) for dynamic update. Click the <b>...</b> button to open the <b>Edit Expression</b> dialog for easier editing of long formulas.

In the **Value List** group it is specified if and how to present a list of values for an input box.

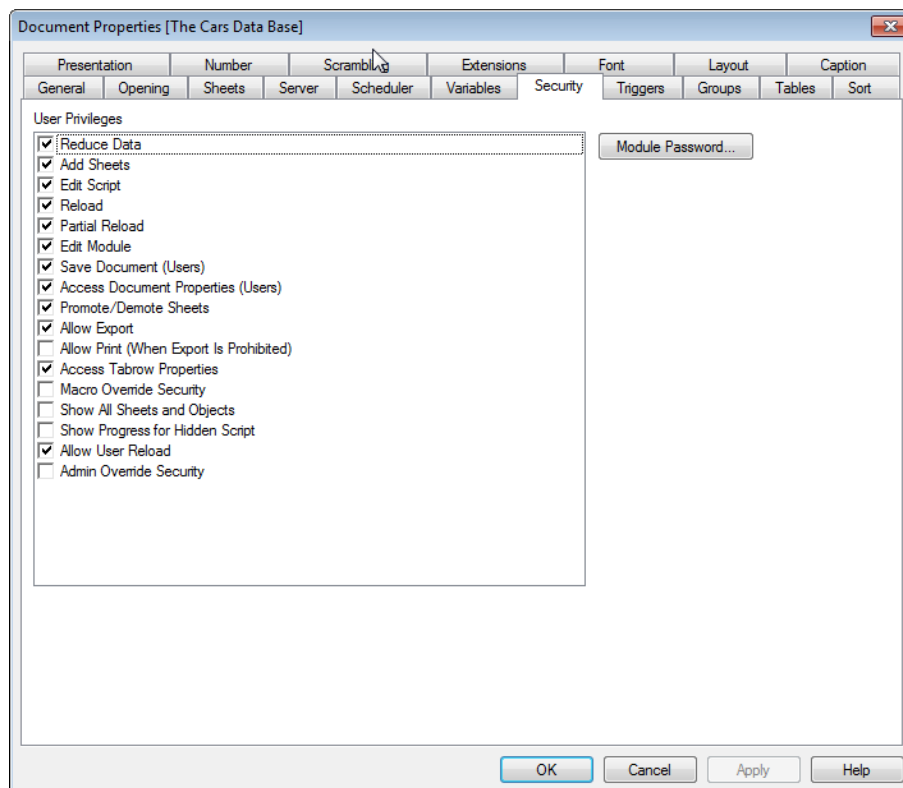
<b>No List</b>	No list of values previously used in the variable is maintained.
----------------	--

- List \_ Recent Values** A drop-down (MRU) list with the values most recently used, 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 **entries** box.
- Predefined Values in Drop-down** A drop-down list with **Predefined Values** 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**.

In the **Predefined Values** group a list of predefined values is specified, which can be presented to the user in a drop down and/or used for defining acceptable variable values.

- Number Series** Check this alternative to generate a list of predefined numeric values based on a lower limit (**From**), an upper limit (**To**) and a **Step** value. This option can be used alone or in combination with **Listed Values**.
- Listed Values** Check this alternative to specify a list of arbitrary predefined values. The values can be either numeric or alphanumeric. Alphanumeric values must be enclosed in quotes (e.g. 'abc'). Each value is separated by a semicolon ( ; ) (e.g. 'abc';45;14.3;'xyz' ). This option can be used alone or in combination with **Number Series**.
- Comment** This is a commentary field where the creator of a variable can describe its purpose and function.

## 30.6 Document Properties: Security



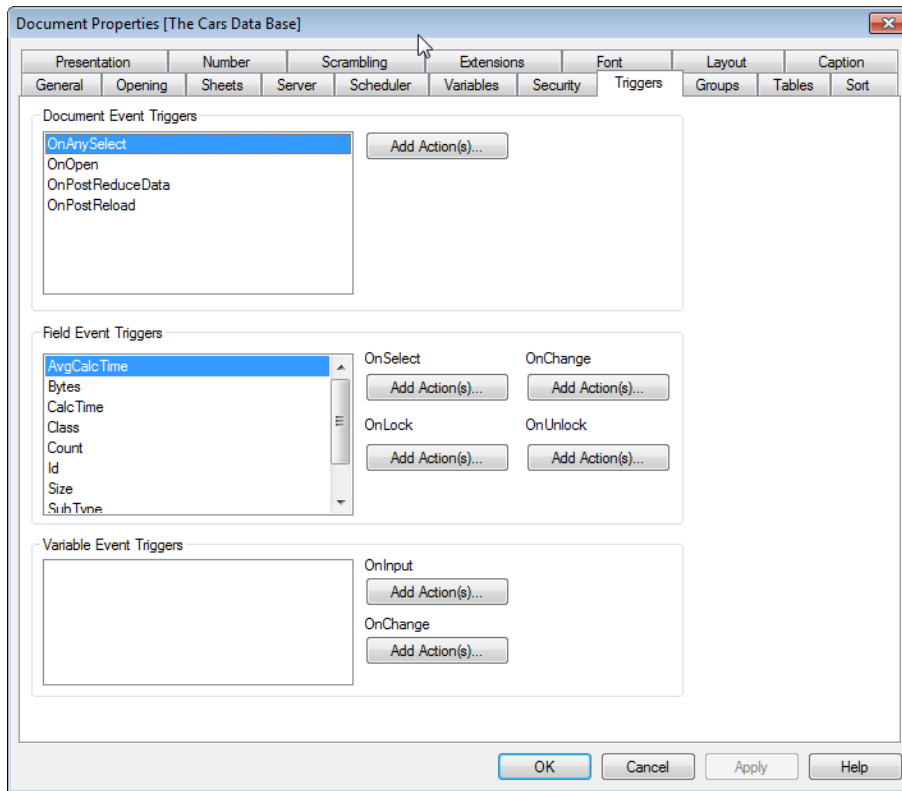
*Document Properties, Security*

In this property tab it is possible to set the **User Privileges** on the document level. By default, all the options are checked (on). People with administrator privileges can always access this dialog and change the settings. The settings can however prevent ordinary users from altering the document in a number of ways. The options are:

<b>Reduce data</b>	Permission to use the command <i>Reduce Data</i> (page 50).
<b>Add Sheets</b>	Permission to add more sheets to the document.
<b>Edit Script</b>	Permission to access the <i>Edit Script Dialog</i> (page 175) dialog.
<b>Reload</b>	Permission to run the load script.
<hr/>	
<b>Note!</b>	
The permissions to <b>Reload</b> and <b>Partial Reload</b> (see below) data are automatically disabled by the operation <b>Reduce Data - Keep Possible Values</b> , which is performed from the <b>File Menu</b> .	
<hr/>	
<b>Partial Reload</b>	Permission to run a <i>Partial Reload</i> (page 50).
<b>Edit Module</b>	Permission to access the <i>Edit Module</i> (page 895) dialog.
<b>Save Document (Users)</b>	Permission to save the document with USER privileges.
<b>Access Document Properties (Users)</b>	Permission to access the <b>Settings: Document Properties</b> with USER privileges.
<b>Promote/Demote Sheets</b>	Permission to use the <b>Promote Sheet</b> and <b>Demote Sheet</b> commands.
<b>Allow Export</b>	Permission to use any <b>Export</b> , <b>Print</b> or <b>Copy to Clipboard</b> commands.
<b>Allow Print (When Export Is Prohibited)</b>	Overrules the <b>Allow Export</b> settings for printing purposes (only).
<b>Access Tabrow Properties</b>	Permission to access the <b>Tabrow Properties</b> .
<b>Macro Override Security</b>	Overrides any security settings by means of macros and commands via automation.
<b>Show All Sheets and Objects</b>	Overrules conditional display of sheets and sheet objects, thus everything becomes visible. This functionality can be toggled by pressing Ctrl+Shift+S.
<b>Show Progress for Hidden Script</b>	Show it in the script progress dialog while executing hidden script.
<b>Allow User Reload</b>	Permission to reload the script when the document is opened in USER mode. When it is deselected, reload is not possible, even if the <b>Reload</b> check box above is selected.
<b>Admin Override Security</b>	Overrides any security settings on document or sheet level by logging in as Administrator.
<b>Module Password</b>	Allowing to set a password for accessing the <i>Edit Module</i> (page 895) Dialog. To change the password, click the <b>Module Password</b> button again.

Further privileges for sheet objects etc. can be set in the *Sheet Properties: Security* (page 467) property page.

## 30.7 Document Properties: Triggers



### *Document Properties, Triggers*

On the **Triggers** tab, it is possible to set actions (including macro actions) to be invoked on document events, variable events and field events. For more information on macros, see the *Edit Module (page 895)*. Not all actions can be invoked from the following event triggers. Read more about the different actions under *Add Action Dialog (page 553)*.

### **Note!**

There are limitations as to the behavior of certain macro triggers when working with documents on Qlik-View Server. Please study the section *Using Macros in QV Documents on the QV-Server (page 903)* before designing server documents with macro triggers.



## Document Event Triggers

In this group actions are set to trigger on selected events in the document. An event has to be selected in the list before it is possible to 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. This event is not supported when running in the AJAX client.

### 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 552)* page. On this page it is possible to add an action to the trigger. Choose the action **macro** to assign an existing macro name or type any name for which a macro can be created later in the *Edit Module (page 895)* dialog.

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 it is possible to change the action for the event.

## Field Event Triggers

In this group actions are set 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 552)* page. Here it is possible to assign an action, including an existing macro name, or type any name for which a macro can be created later in the *Edit Module (page 895)* dialog. For more information about adding actions, see *Actions (page 552)*.

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 this 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 552)* 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. For more information about adding actions, see *Actions (page 552)*.

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.

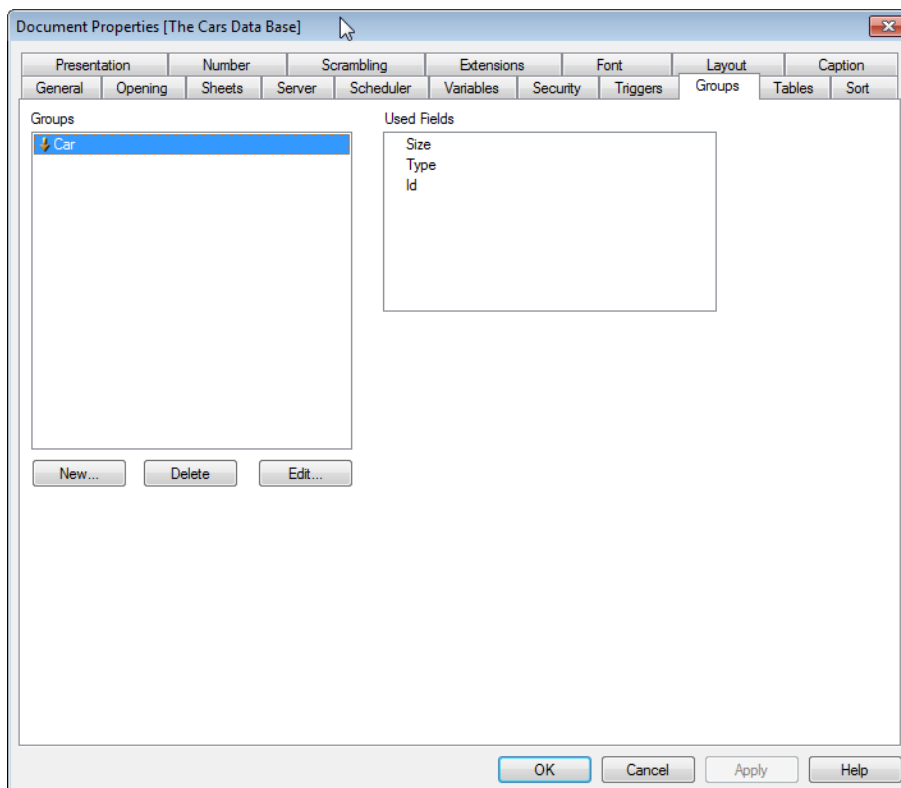
---

#### Note!

Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

---

## 30.8 Document Properties: Groups



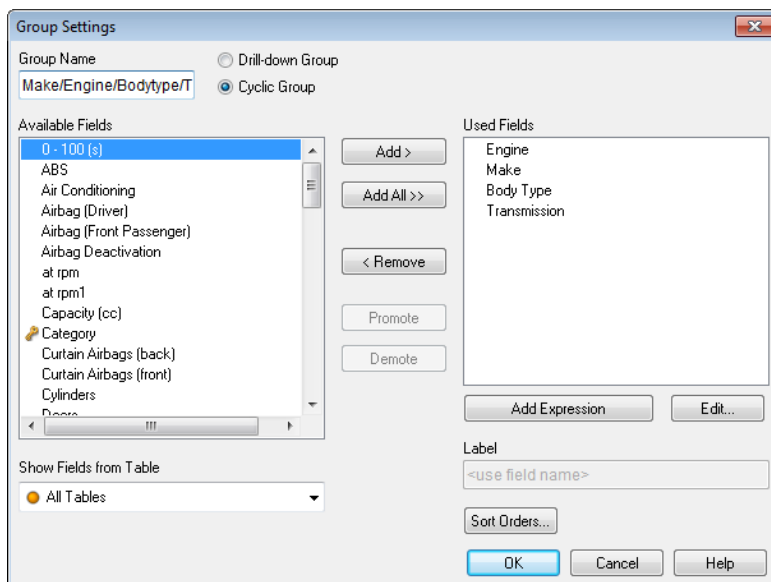
*Document Properties, Groups***Note!**

In order for the **Groups** tab to be available the document must contain data and the script must have been reloaded once.

In this tab it is possible to create drill-down or cyclic field groups. For more information on how to use groups, see *Non-Hierarchical Groups (Cyclic)* (page 874) and *Hierarchical Groups (Drill-Down)* (page 873). 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.

- |               |  |
|---------------|--|
| <b>New</b>    | Clicking this button opens the <i>Group Settings</i> (page 443) dialog where it is possible to define a new field group. |
| <b>Delete</b> | Deletes the selected group.  |
| <b>Edit</b>   | Opens the <i>Group Settings</i> (page 443) dialog for the selected group. This page is used for accessing field groups.  |

## Group Settings

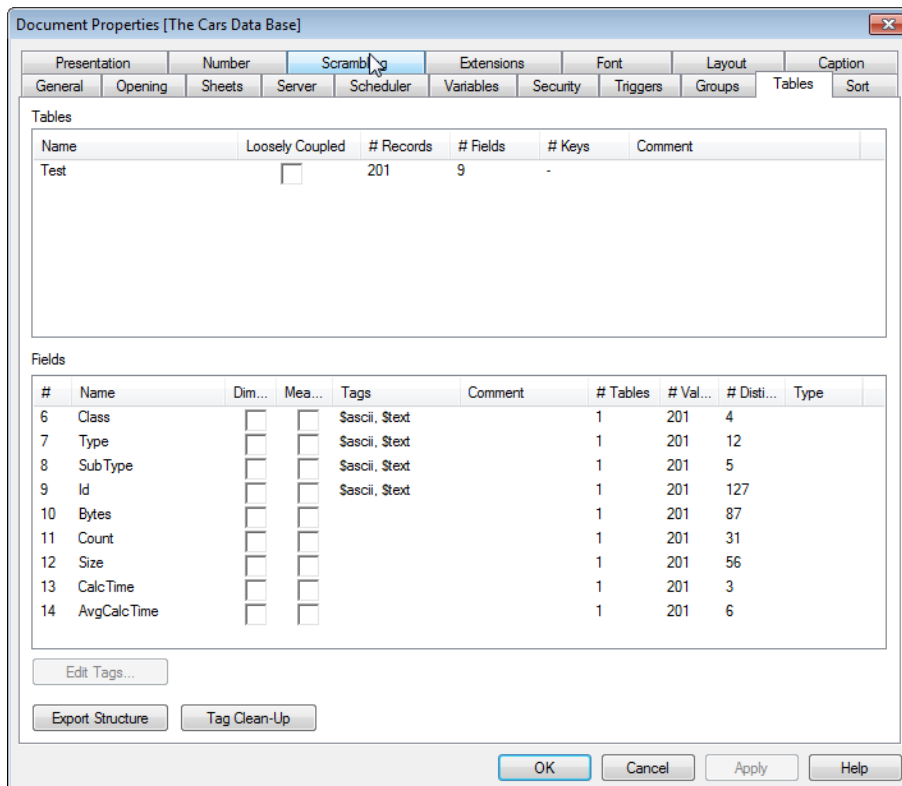
*The Group Settings dialog*

This dialog is opened when clicking the **New...** or **Edit...** buttons in the *Document Properties: Groups* (page 442) dialog.

- |                               |   |
|-------------------------------|---|
| <b>Group Name</b>             | Enter a name for the group and select either <b>Drill-down Group</b> or <b>Cyclic Group</b> .<br><hr/> <b>Note:</b> Do not use a name identical to the name of any field that is used in the group. |
| <b>Available Fields</b>       | Available fields are listed in this column. Select the fields to include in the group.  |
| <b>Show Fields from Table</b> | Select a table.   |
| <b>Add</b>                    | Moves the selected fields from the column of <b>Available Fields</b> to the column of <b>Used Fields</b> .  |

- Add All** Moves all fields from 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**.
- 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**.
- Used Fields** Fields included in the current group are listed in this column. Initially the column is empty.
- Add Expression** Opens the *Edit Expression Dialog (page 757)* dialog for easy editing of long formulas.
- Edit...** Opens the *Edit Expression Dialog (page 757)* 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 (page 871)* for dynamic update.
- Sort Orders...** Clicking on this button opens the **Group Sort Order** dialog in which a separate sort order can be set for each field in the group.

## 30.9 Document Properties: Tables



*Document Properties, Tables*

When data that includes circular references is loaded into QlikView, loosely coupled tables are created automatically, to avoid that the circular references create a loop in the QlikView internal logic. These loosely coupled tables need to be handled in order to visualize data in a way that is expected and understandable. See *Circular References (page 111)* for further information.

Any table can also be made loosely coupled interactively from this dialog or via macros. Additionally, it is possible to declare loosely coupled tables explicitly in the script via the *Loosen Table (page 260)* statement. The normal QlikView associative logic is disconnected internally for loosely coupled tables. This means that selections in one field do not propagate through to the other fields in the table. This may be very useful in a number of situations, typically when it is needed to avoid circular references in the data structure. For more examples of how this feature can be used, please refer to the section *Intentionally Creating Loosely Coupled Tables (page 925)*.

---

**Note!**

Making one or more tables loosely coupled can radically change the behavior of the document. Use this feature only when you are absolutely sure about what you are doing!

---

This dialog page consists of two lists, the **Tables** list and the **Fields** list. Either table can be sorted on any column, by clicking its column header.

The **Tables** list contains the following information:

<b>Name</b>	The name of the internal table.
<b>Loosely Coupled</b>	If the alternative is checked, the table is loosely coupled. It is possible to deliberately set this alternative for a table here.
<b># Records</b>	The number of records (rows) of the table.
<b># Fields</b>	The number of fields (columns) of the table.
<b># Keys</b>	The number of key (connecting) fields (columns) of the table.
<b>Comment</b>	Displays the comments read from the data source and the <i>Comment Field (page 226)</i> made on the field.

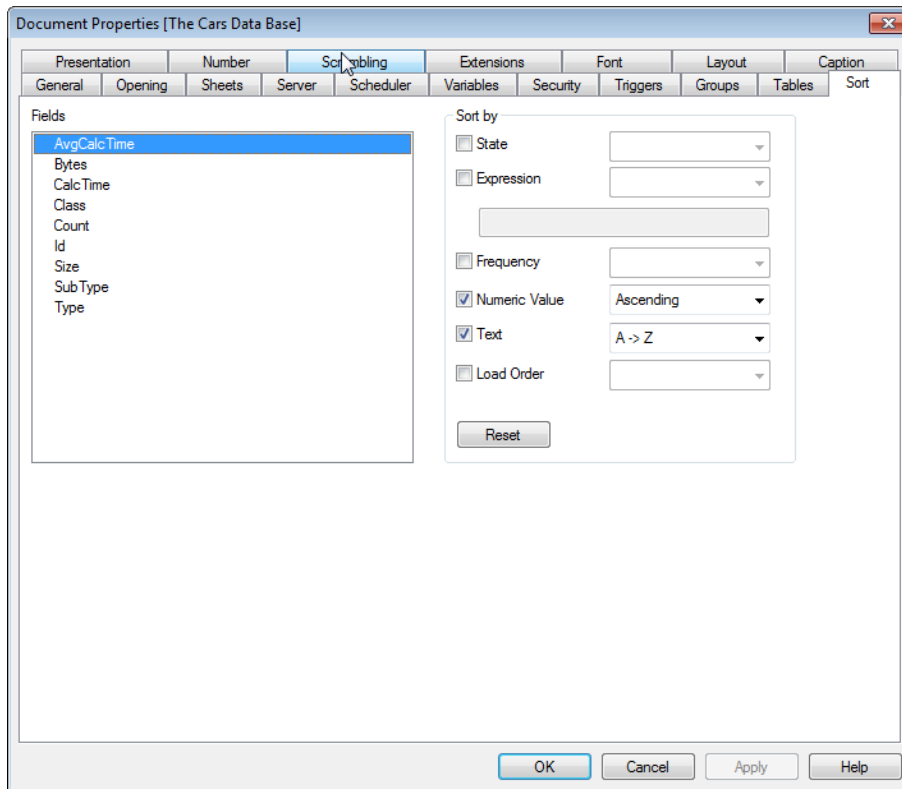
The **Fields** list at the bottom of the page shows all fields of the QlikView document or, if one internal table has been selected in the list above, the fields of that table. The columns are as follows:

<b>#</b>	The internal number of the field. Numbers 0 to 5 are used by the QlikView system fields, which are not displayed in this list.
<b>Name</b>	The name of the field.
<b>Dimensions</b>	Mark the check box to the right of the field name to add the system tag \$dimension to the field. This tag denotes a field recommended for use in chart dimensions, list boxes etc. A field tagged with <i>dimension</i> will be displayed at the top of all field selection controls in QlikView except in the <b>Edit Expression</b> dialog.
<b>Measures</b>	Mark the check box to the right of the field name to add the system tag \$measure to the field. This tag denotes a field recommended for use in expressions. A field tagged with <i>measure</i> will be displayed at the top of all field selection controls in the <b>Edit Expression</b> dialog.
<b>Tags</b>	Displays the <i>Field Tags (page 171)</i> . \$ denotes a system tag.

---

<b>Comment</b>	Displays the comments read from the data source and an indicator for any special status of the field, such as <b>Semantic</b> , <b>AndMode</b> , <b>AlwaysOneSelected</b> , <b>Info</b> , <b>Locked</b> or <b>Hidden</b> .
<b># Tables</b>	The number of tables in which the field occurs.
<b># Values</b>	The total number of field values, disregarding selections. This information is not available for key (connecting) fields.
<b># Distinct</b>	The total number of distinct field values, disregarding selections.
<b>Type</b>	Displays an indicator for any special status of the field, such as <b>Semantic</b> , <b>AndMode</b> , <b>AlwaysOneSelected</b> , <b>Locked</b> or <b>Hidden</b> . See also: <i>System Variables (page 283)</i> .
<b>Edit Tags...</b>	Opens a dialog where it is possible to add and remove tags. System tags cannot be removed here. Tags added may not be given a name occupied by a system tag.
<b>Export Structure</b>	By pressing this button it is possible to 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 the target folder can be chosen for the export. The default is to put the files in the same folder as the QlikView document.
<b>Tag Clean-Up</b>	Click this button to clean up any tags that might be left after a field has been removed from the QlikView document.

## 30.10 Document Properties: Sort



### *Document Properties, Sort*

On this property tab the sort order for the values of any field in the document can be configured. (The sort order of a field can also be set from *List Box Properties: Sort* (page 486).)

Select a field in the **Fields** list to the left, then check one or more of the options to the right. The options are described below.

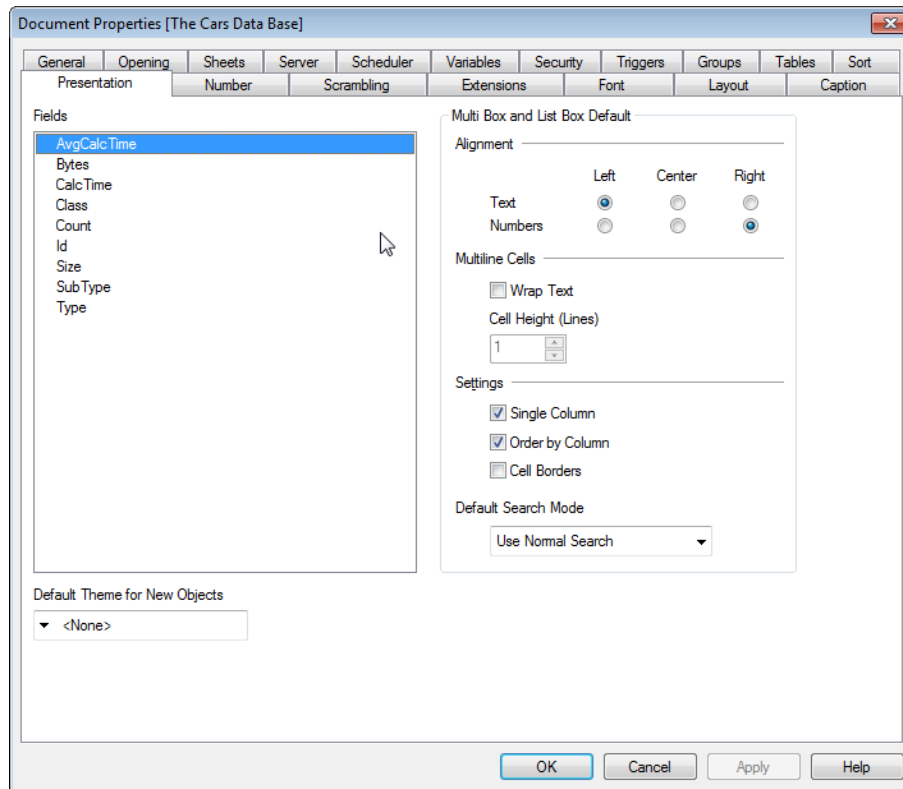
In the **Sort by** group the default sort order of the field values in the sheet objects can be set. Changes made in this group will affect fields in sheet objects created after the change. Previously created sheet objects will not be affected.

<b>State</b>	Toggles the status whether the sorting procedure of values should be made by selection state. The selected values appear at the top of the list if this option is set. The sort order is: <b>selected, possible, excluded</b> (ascending).
<b>Expression</b>	Field values will be sorted according to an arbitrary expression that is entered into the text edit box below this sort option.
<b>Frequency</b>	Toggles the status whether the sorting procedure of values should be made by frequency.
<b>Numeric Value</b>	Toggles the status whether the sorting procedure of values should be made by numeric value.
<b>Text</b>	Toggles the status whether the sorting procedure of values should be made in alphabetical order according to the ASCII standard.
<b>Load Order</b>	Toggles the status whether the sorting procedure of values should be made in load order.

The **Reset** button sort order is sort by state, ascending and sort by Text, A --> Z.

If more than one sort order is specified the sort order will be state, expression, frequency, numeric, text, load order.

## 30.11 Document Properties: Presentation



### *Document Properties, Presentation*

This property tab makes it possible to specify the default presentation settings of the field values used for subsequently created list boxes and multi boxes. The available fields are listed in the **Fields** group.

The **Multi Box and List Box Default** group contains the sub-groups **Alignment**, **Multiline Cells** and **Settings**. **Default Search Mode** specifies the initial default search mode to be used in text search.

#### **Default Theme for New Objects**

A QlikView theme can be selected here which will 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 Qlik-View document.

At the bottom of the drop-down list there is a Browse... command in case the theme file 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.

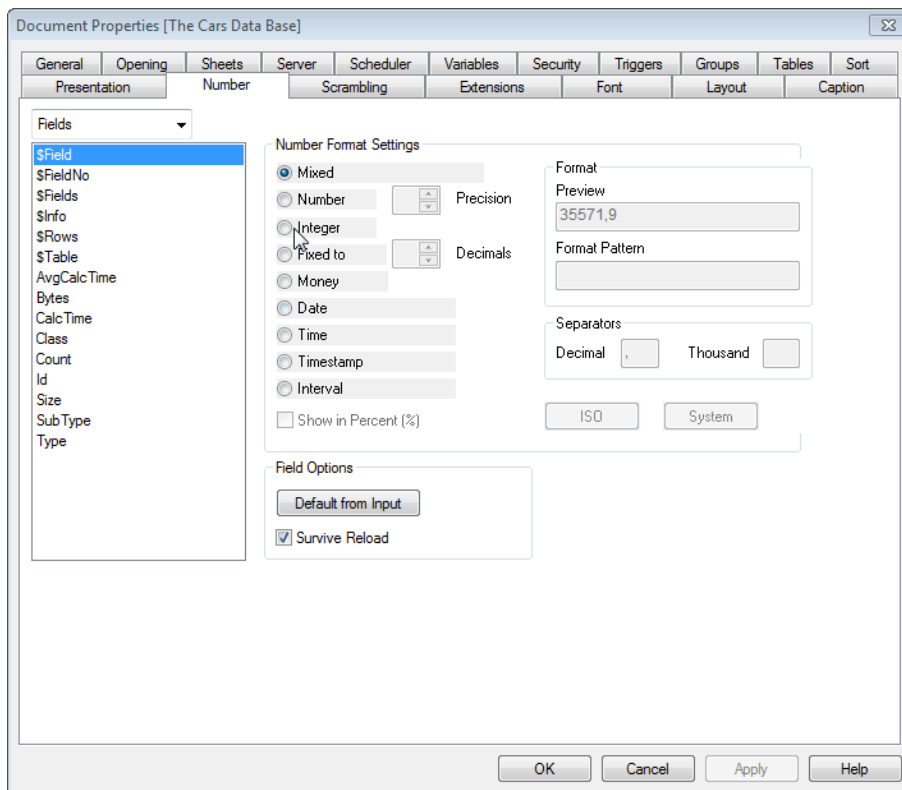
#### **Alignment**

The default alignment of field values can be set separately for **Text** and **Numbers**. As default, text values are left aligned, numbers values are right aligned.



- Multiline Cells** If the **Wrap Text** option is checked, the (text) contents of the cell (field value) will be displayed in two or more lines.  
The number of cell lines is set in the **Cell Height ( Lines )** edit box.
- Settings**
- Single Column**  
Checking this option forces single column mode.
- Order by Column**  
In multiple column displays the field values will be ordered by column, rather than by row.
- Cell Borders**  
Checking this option will introduce horizontal lines between field values.
- Default Search Mode**
- Use Previous**  
The search mode of the last completed search will be used.
- Use Wildcard Search**  
The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search.
- Use Normal Search**  
No additional characters will be added to the search string. Without wildcards, a Normal Search will be made.

## 30.12 Document Properties: Number



### *Document Properties, Number*

This property tab provides number formatting settings for all fields and variables of the document. The list on the left is set to display either the **Fields** or the **Variables** of the document.

The **Field Options** group is only used for **Fields**:

The **Default from Input** button sets the formatting to that of the input.

Check the **Survive Reload** option if the formatting is to apply to the document permanently.

The following controls for formatting values are available:

<b>Mixed</b>	Both numbers and text. Numbers shown in original format.
<b>Number</b>	Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.
<b>Integer</b>	Shows numeric values as integers.
<b>Fixed to</b>	Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.
<b>Money</b>	Shows numeric values in the format shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.
<b>Date</b>	Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Time</b>	Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Timestamp</b>	Shows values that can be interpreted as date + time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Interval</b>	Shows time as sequential time increment (e.g. format = <i>mm</i> shows the value as the number of minutes since calendar start ( 1899:12:30:24:00).

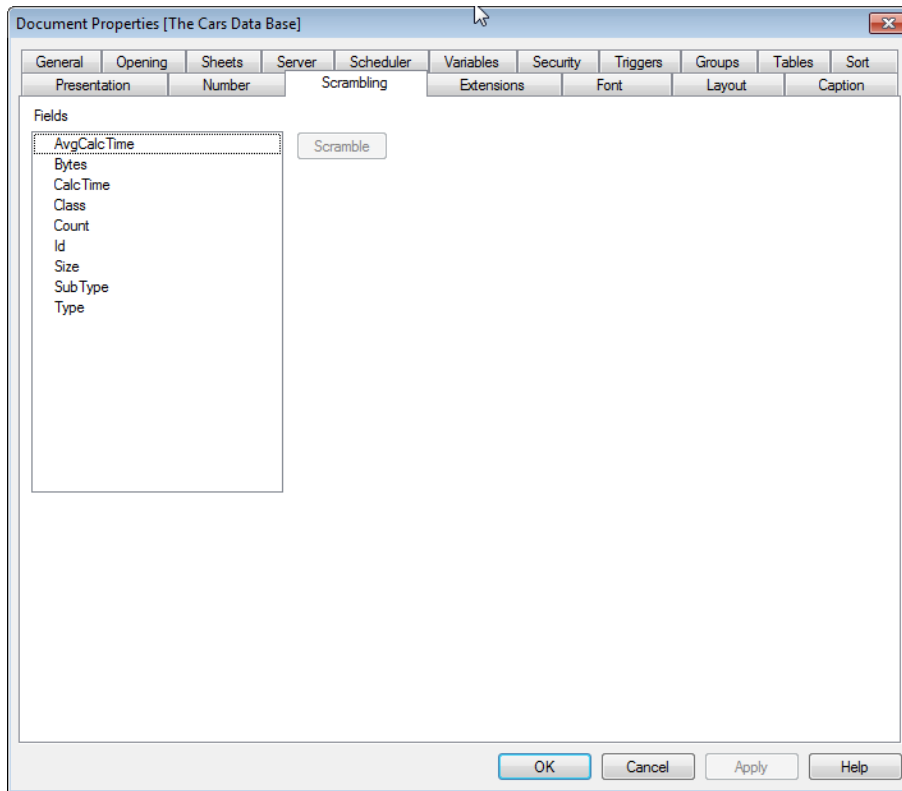
The **Show in Percent (%)** button operates on the following formats: **Number**, **Integer** and **Fixed to**.

**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

The **ISO** button uses the ISO standard to format date, time and timestamp values.

The **System** button uses the system settings for the formatting.

## 30.13 Document Properties: Scrambling

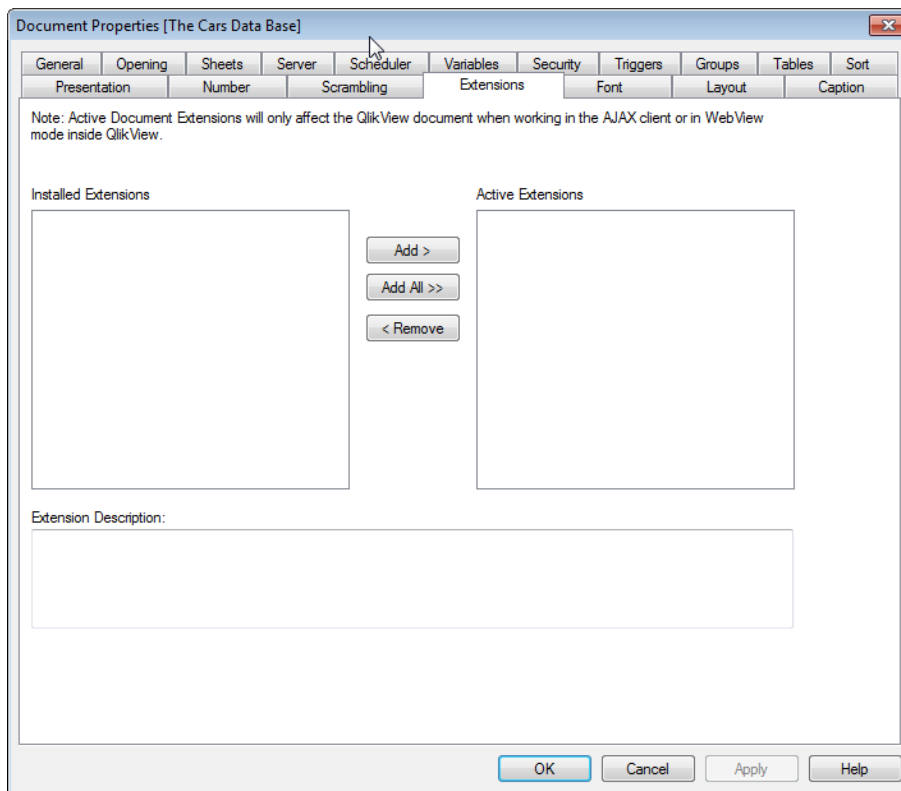


### *Document Properties, Scrambling*

This tab 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.

- |                 |   |
|-----------------|---|
| <b>Fields</b>   | This is a list of all fields in the document. Select one or more fields for scrambling.   |
| <b>Scramble</b> | 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 the script is re-executed, however, the scrambling will be lost. |

## 30.14 Document Properties: Extensions



### *Document Properties, Extensions*

In this tab it is possible to select extensions to alter the appearance of the document.

#### **Note!**

Active Document Extensions will only affect the QlikView document when working in the AJAX client or in WebView mode inside QlikView.

#### **Note!**

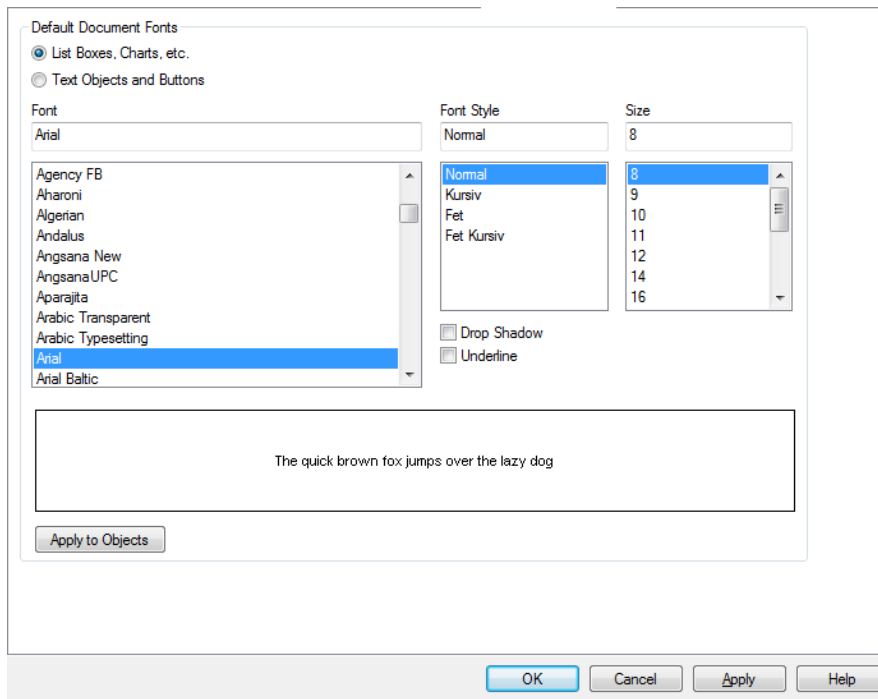
Document Extensions are created by the application designer/developer. Document Extensions provide a mechanism for injecting JavaScript code into a QlikView application viewed through the AJAX client. Variations in the Extensions can occur on web browsers with different JavaScript engines.

The **Installed Extensions** list shows all installed extensions. An extension can be selected by clicking in the list. The selected extensions are shown in the **Active Extensions** area.

<b>Installed Extensions</b>	A list of installed extensions are shown here.
<b>Add</b>	Moves the selected fields in the column of <b>Installed Extensions</b> to the column of <b>Active Extensions</b> .
<b>Add All</b>	Moves all fields from the column of <b>Installed Extensions</b> to the column of <b>Active Extensions</b> .
<b>Remove</b>	Moves the selected fields in the column of <b>Active Extensions</b> to the column of <b>Installed Extensions</b> .
<b>Active Extensions</b>	A list of active extensions are shown here. These are the extensions added from the column of <b>Installed Extensions</b> .

**Extension Description** A short description about the extension, highlighted in one of the columns, is displayed here.

## 30.15 Font



### *The Font dialog*

Here the **Font**, **Font style** and **Size** of the font to be used can be set.

The font can be set for any single object (**Object Properties: Font**), or all objects in a document (**Apply to Objects on Document Properties: Font**).

Further, the default document fonts for new objects can be set on **Document Properties: Font**. There are two default fonts:

1. The first default font (**List Boxes, Charts, etc**) is used for most objects, including list boxes and charts.
2. The second default font (**Text Objects and Buttons**) is used for buttons and text boxes, which are objects that usually need a larger font.

Finally, the default fonts for new documents can be set on **User Preferences: Font**.

For charts, buttons and text objects (except search objects) a font **Color** can also be specified. The color may be **Fixed** (Click the colored button to specify a different color) or it can be dynamically **Calculated** from an expression. The expression must be a valid color representation, which is created using the *Color Functions* (page 375). If the result of the expression is not a valid color representation, the font color will default to black.

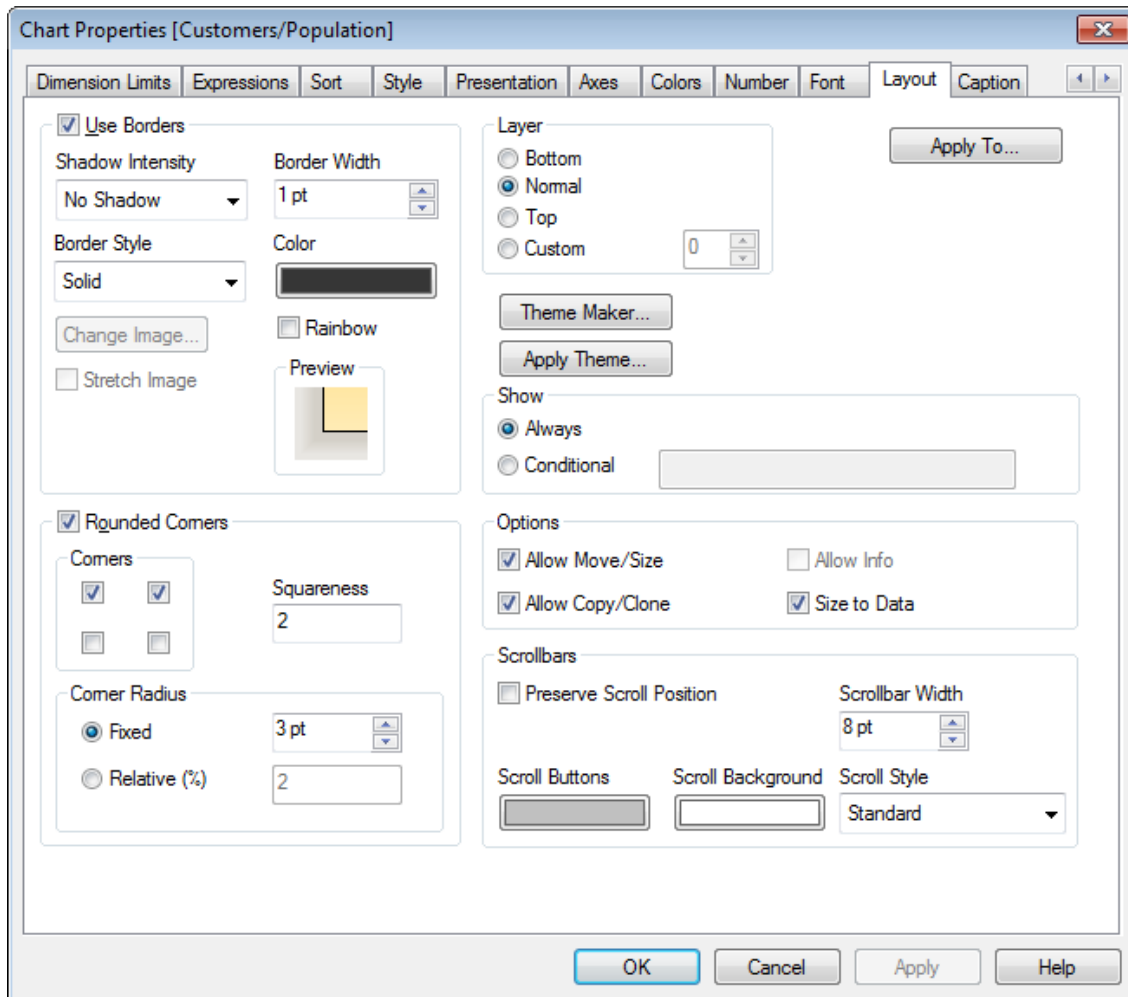
Additional settings are:

**Drop Shadow** If this option is checked a drop shadow will be added to the text.

**Underline** If this option is checked the text will be underlined.

A sample of the selected font is shown in the preview pane.

## 30.16 Layout



### *The Layout page*

A Layout setting will apply to the current object only, if it is made from the Object Properties page.

A Layout setting will apply to all objects of the specified type(s) in the document, if it is made from the Document Properties page.

### **Use Borders**

Enable this setting in order to use a border around the sheet object. Specify the type of border by selecting it in the drop-down menu.

#### **Shadow Intensity**

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**.

<b>Border Style</b>	The following predefined border types are available: <b>Solid</b> A solid unicolored border. <b>Depressed</b> Border giving the impression of depressing the sheet object from the background. <b>Raised</b> Border giving the impression of raising the sheet object from the background. <b>Walled</b> Border giving the impression of a wall around the sheet object. <b>Image</b> Border specified by custom image.
---------------------	---

**Change Image...** If the **Image** border type is selected, 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.

**Border Width** This option is 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).

**Color** Click this button to open a dialog in which an appropriate base color can be chosen 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 chosen in *Document Properties: General (page 424)*, there is no choice of border type, there is only the **Shadow Intensity** drop-down menu and the **Border Width** setting.

### **Rounded Corners**

In the **Rounded Corners** group the general shape of the sheet object is defined. These settings allow for drawing of sheet objects ranging from perfectly circular/elliptical via super elliptical to rectangular:

**Rounded Corners** Checking this option makes the alternatives for rounded corner shapes possible.

**Corners** Corners for which the check box remains unmarked will be drawn rectangular instead.

**Squareness** A variable number between 2 and 100 where 100 defines a rectangle with perfectly square corners and 2 corresponds to a perfect ellipse (a circle for a 1:1 aspect ratio). A squareness between 2 and 5 is usually optimal for achieving rounded corners.

**Corner Radius** This setting determines the radius of the corners in fixed distance (**Fixed**) or in percent of the total quadrant (**Relative (%)**). This setting lets you control the extent to which the corners will be affected by the underlying general shape set under **Squareness**. The 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 an object can be defined as residing in one of three layers:

<b>Bottom</b>	A sheet object with the <b>Bottom</b> layer property can never obscure sheet objects in the <b>Normal</b> and <b>Top</b> layers. It can only be placed on top of other sheet objects in the <b>Bottom</b> layer.
<b>Normal</b>	When created, sheet objects reside in the <b>Normal</b> (middle) layer. A sheet object in the <b>Normal</b> layer can never be obscured by sheet objects in the <b>Bottom</b> layer and can never obscure sheet objects in the <b>Top</b> layer.
<b>Top</b>	A sheet object in the <b>Top</b> layer can never be obscured by sheet objects in the <b>Normal</b> and <b>Bottom</b> layers. Only other sheet objects in the <b>Top</b> layer can be placed on top of it.
<b>Custom</b>	The <b>Top</b> , <b>Normal</b> and <b>Bottom</b> 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.

### Show

In the **Show** group it is possible to specify a condition under which the sheet object is displayed:

<b>Always</b>	The sheet object will always be displayed.
<b>Conditional</b>	The sheet object will be shown or hidden depending on a <i>Conditional Functions (page 340)</i> which will be evaluated continuously depending on e.g. selections etc. The sheet object will only be visible when the condition returns TRUE.

---

### Note!

Users with Admin privileges for the document can override all show conditions with the **Show All Sheets and Sheet Objects** in the *Document Properties: Security (page 438)*. This functionality can be toggled by pressing Ctrl+Shift+S.

---

### Options

In the **Options** group, it is possible to disallow moving and resizing of the sheet object. The settings in this group are only relevant if the corresponding check boxes are enabled in **Document Properties: Layout** and **Sheet Properties: Security**.

<b>Allow Move/Size</b>	If this option has been deselected it will be impossible to move or resize the sheet object.
<b>Allow Copy/Clone</b>	If this option has been deselected it will be impossible to make a copy of the sheet object.
<b>Allow Info</b>	When the <i>Info (page 245)</i> function is in use an info icon will be displayed in the window caption whenever a field value has information associated to it. If you don't want the info icon to be displayed in the caption, you can uncheck this option.
<b>Size to Data</b>	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

Various controls for changing the scrollbar layout are located in the **Scrollbars** group:

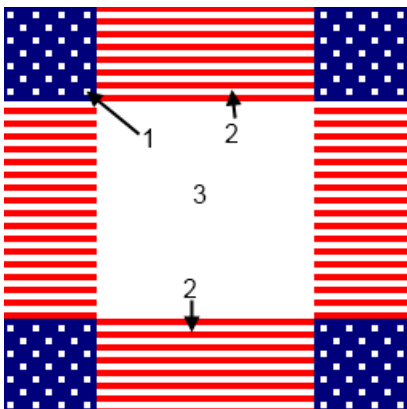


<b>Preserve Scroll Position</b>	With this setting enabled, QlikView will try to preserve the vertical scroll position of tables and charts with an x-axis scroll bar when a selection is made in another object. The setting must be enabled in <i>User Preferences: Objects (page 78)</i> as well.
<b>Scroll Buttons</b>	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. Either color can be defined as a solid color or a gradient via the <i>Color Area (page 428)</i> dialog that opens when you click the appropriate button.
<b>Scrollbar Width</b>	This control affects both the width and the relative size of the scrollbar symbols.
<b>Scroll Style</b>	Sets the scroll bar style. Select a style in the drop-down control. The <b>Classic</b> scroll bar style corresponds to QlikView 4/5 scroll bars. The <b>Standard</b> scroll bar style gives a more modern look. The third style is <b>Light</b> , which is a thinner, lighter bar.
<b>Apply To...</b>	Opens the <b>Caption and Border Properties</b> dialog where you can set where to apply the properties that have been set on the <b>Layout</b> page.

## Example of an Image Border Definition File

An image border is defined by a single image file containing the following elements:

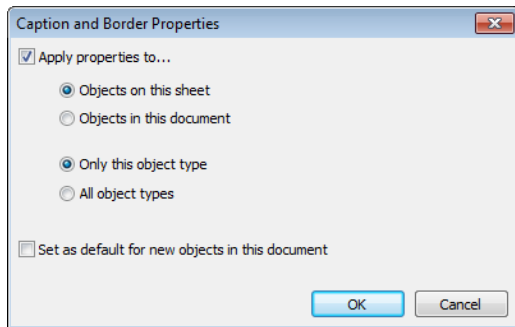
1. The four corners must form a square of pixels. In the case below each corner is 15 pixels by 15 pixels. The corners may be identical or different, but must all be drawn separately.
2. The areas between the corners describe the top, bottom, left and right borders between the corners. It is only needed 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 (page 493)*.
3. The area in the middle has no function and may contain anything or nothing.



### *The Width Box*

The number of pixels in the corner square must be correctly stated in the **Width Box** 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.

## Caption and Border Properties



*The Caption and Border properties page*

In this dialog you set to which objects in the document the caption and border properties should apply.

### **Apply properties to...**

Mark this check box to apply the settings to other objects than the current one.

1. Choose one of the options:

#### **Objects on this sheet**

Applies the 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 the settings to objects of the entire document.

2. Choose one of the options:

#### **Only this object type**

Applies the 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 the settings to all objects.

### **Set as default for new objects in this document**

Mark this check box to use the 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.

## 30.17 Caption

### *The Caption page*

A Caption setting will apply to the current object only, if it is made from the **Object Properties** page. A Caption setting will apply to all objects of the specified type(s) in the document, if it is made from the **Document Properties** page.

On the **Caption** page, specify layout options that are completely different from the general layout of the object.

**Show Caption** When this option has been checked a caption will be drawn at the top of the sheet object. List boxes and other "box objects" will have the option switched on by default whereas buttons, text objects and line/arrow objects will not.

**Title Text** In the text box you can enter a title to be shown in the caption of the sheet object. Use the **Font...** button to change the caption font.

Set the colors of the caption in its different states. The settings for **Active Colors** and **Inactive Colors** can be made separately from each other.

Click the **Background Color** or **Text Color** button to open the *Color Area (page 428)* dialog. The **Background Color** can be defined as a **solid** or a **gradient** color on the **Color Area** dialog. **Text Color** can be defined a **Fixed** or a **Calculated** color using *Color Functions (page 375)*.

**Wrap Text** If this option is checked, the caption will be displayed in two or more rows.

**Caption Height (Lines)** Set the number of caption lines in this edit box.

The precise size and position of the QlikView object can be determined and adjusted by the size/position settings for the **Normal** or **Minimized** QlikView sheet object. These settings are measured in pixels:

**X-pos** Sets the horizontal position of the left side of sheet object in relation to the sheet's left edge.

<b>Y-pos</b>	Sets the vertical position of the top side of sheet object in relation to the sheet's upper edge.
<b>Width</b>	Sets the width of the QlikView sheet object.
<b>Height</b>	Sets the height of the QlikView sheet object.

The orientation of the caption label can be altered with the **Caption Alignment** options:

<b>Horizontal</b>	The label can be horizontally aligned: <b>Left</b> , <b>Centered</b> or <b>Right</b> within the caption area.
<b>Vertical</b>	The label can be vertically aligned: <b>Top</b> , <b>Centered</b> or <b>Bottom</b> within the caption area.

### 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.

---

#### Note!

Use the special caption icons with care. Too many icons will only confuse the user.

---

<b>Allow Minimize</b>	When this option has been checked a minimize icon will be displayed in the window caption of the sheet object, providing that the object is minimizable. Furthermore, this will make it possible to minimize the object by double-clicking the caption.
<b>Auto Minimize</b>	This option becomes available when <b>Allow Minimize</b> is checked. When <b>Auto Minimize</b> is checked for several sheet objects on the same sheet, all but one will be automatically minimized at any time. This is useful e.g. for alternately displaying several graphs in the same sheet area.
<b>Allow Maximize</b>	When this option has been checked a maximize icon will be displayed in the window caption of the sheet object, providing that the object is maximizable. Furthermore, this will make it possible to maximizing the object by double-clicking the caption. If both <b>Allow Minimize</b> and <b>Allow Maximize</b> are checked, double-clicking will effect minimizing of the object.
<b>Help Text</b>	Here you may enter a help text to be displayed in a pop-up window. The help text may be specified as a <i>Calculated Formula (page 871)</i> . This option is not available at document level. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas. Enter e.g. a description of the sheet object. A help icon will be added to the window caption of the object. When the mouse pointer is over the icon, the text will be displayed in a pop-up window.

# 31 The Sheet

A QlikView document can have one or several sheets on which 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.

## 31.1 Creation

New sheets are created by choosing **Add Sheet** from the **Layout** menu or by clicking the **Add Sheet** button in the toolbar.

## 31.2 Navigation

All sheets have tabs attached to them on which the sheet name is printed. By clicking on a tab, the sheet attached to it is activated. 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:

<b>Tabrow Properties...</b>	Opens a dialog in which you can choose a font for the tab. Only Truetype fonts may be used. See <i>Tabrow Properties (page 469)</i> .
<b>Sheet Properties...</b>	Selects the sheet and opens the <b>Sheet Properties</b> 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.
<b>Copy Sheet</b>	Makes a complete copy of the entire sheet with all its sheet objects. The copy will be named "Copy of <i>Sheetname</i> " and positioned as the last sheet in the document.
<b>Promote Sheet</b>	Moves the sheet one step further to the left.
<b>Demote Sheet</b>	Moves the sheet one step further to the right.
<b>Help</b>	Opens context-specific help.
<b>Remove</b>	Makes the sheet active, then removes it.

If you click the tab of active sheet, 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.

## 31.3 Sheet: Object Menu

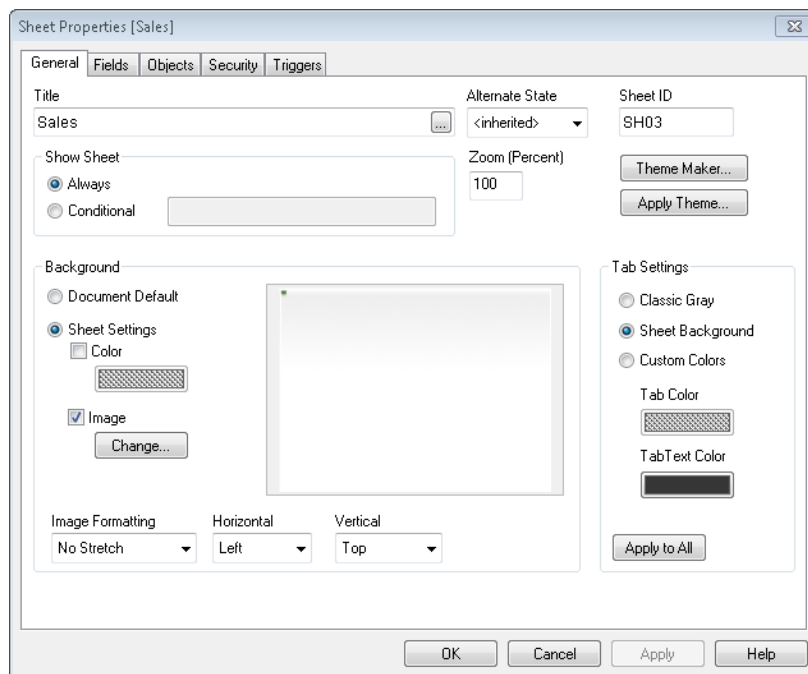
<b>Properties</b>	Opens the <b>Sheet Properties</b> dialog page where you can set the parameters defining the sheet.
<b>Select Fields</b>	Opens the <b>Sheet Properties: Fields</b> page where you can select a field or more to be displayed as a list box on the sheet.
<b>New Sheet Object</b>	Opens a cascading menu listing sheet objects.
<b>Copy Sheet</b>	Makes a complete copy of the entire sheet with all its sheet objects. The copy will be named "Copy of <i>Sheetname</i> " and positioned as the last sheet in the document.
<b>Paste Sheet Object</b>	Pastes a sheet object, previously copied to clipboard, on the sheet. This command can also be invoked by the following keyboard shortcut: Ctrl+V.
<b>Paste Sheet Object as Link</b>	Pastes a linked object, previously copied to the clipboard, on the sheet. The object is directly linked to the original, sharing all attributes and object ID.
<b>Print</b>	<b>Print</b> will result in the printout of an image of the sheet area including any sheet objects.
<b>Copy Image to Clipboard</b>	Copies a bitmap picture of the sheet area (only) to <b>Clipboard</b> .
<b>Export Image to File</b>	Opens a dialog for saving an image of the current sheet to a file. The image can be saved as bmp, jpg, gif or png.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the sheet and all its sheet objects.

## 31.4 Sheet Properties

Opens the *Sheet Properties: General (page 463)* dialog, where sheet specific parameters such as sheet background, fields to display, default font and object layout etc. can be set.

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 *Sheet Properties: Security (page 467)*).

## 31.5 Sheet Properties: General



*Sheet Properties, General*

To name the sheet, enter text in the **Title** edit box. The name will be displayed on the sheet tab.

### **Alternate State**

Choose one of the available states in the list. The following Alternate States are always available.

#### **Inherited**

The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.

#### **Default state**

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

### **Sheet ID**

This is used for *Internal Macro Interpreter* (page 893) purposes. Every sheet is assigned a unique ID, starting with SH01. It is possible to edit this ID number later on.

In the **Show Sheet** group it is possible to specify a condition under which the sheet is displayed:

### **Always**

The sheet will always be displayed.

**Conditional** The sheet will be shown or hidden depending on a conditional expression which will be evaluated continuously depending on e.g. selections etc. The sheet will only be visible when the condition returns TRUE.  
When all sheets in a document are disabled due to **Show Sheet** restrictions a message "No sheets available" will be displayed. Users with Admin privileges for the document can override this condition by checking **Show All Sheets and Sheet Objects** in the *Document Properties: Security (page 438)* page. This functionality can be toggled by pressing Ctrl+Shift+S.

With the **Zoom (Percent)** option you can conveniently alter the size of the displayed active sheet and everything on it.

In the **Background** group you may define a background color for the sheet. The alternative **Document Default** applies the background color defined in the **Document Properties** dialog. Checking **Sheet Settings** and clicking the **Colored** button lets you define a custom background color which can be defined as a solid color or a gradient via the **Color Area** dialog.

Alternatively, the background of the sheet can be customized by checking **Image** and clicking the **Change** button. This will open the **Select Image** dialog where a file with the picture to be used can be selected.

**Image Formatting:**

**No Stretch** The image is displayed as is, which may cause masking problems.  
**Fill** The image is stretched in order to fill the sheet, without regard to aspect ratio.  
**Keep Aspect** The image is stretched as far as possible while retaining the correct aspect ratio.  
**Fill with Aspect** The image is stretched as far as possible while retaining the correct aspect ratio. Areas not covered are then filled by cropping of the image.  
**Tiled** If this option is selected, the image will be tiled the as many times as space allows.

**Horizontal and Vertical orientation:**

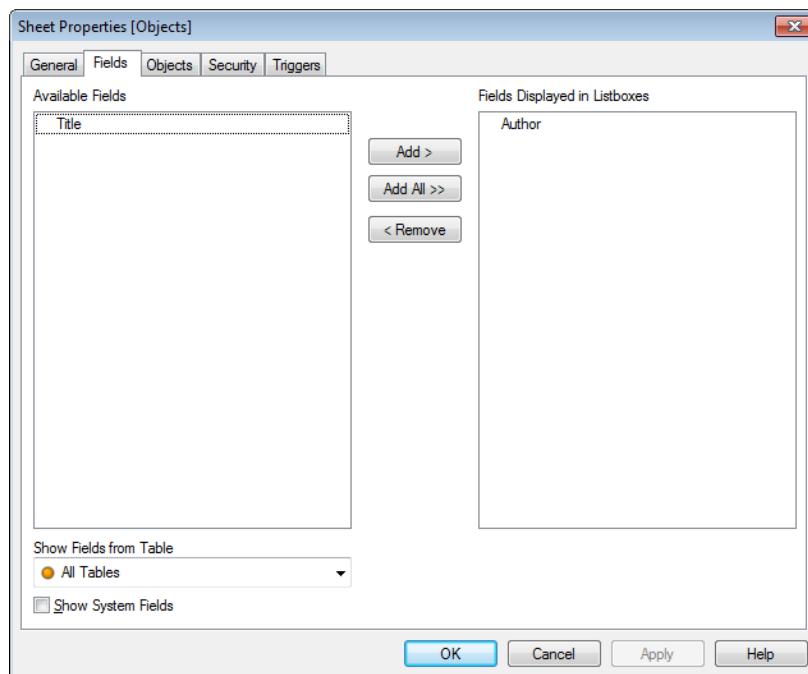
**Horizontal** The image can be horizontally aligned: **Left**, **Centered** or **Right**.  
**Vertical** The image can be vertically aligned: **Top**, **Centered** or **Bottom**.  
**Theme Maker...** Opens the *Theme Maker Wizard (page 603)* for creating a new theme or editing an existing one. See *Layout Themes (page 601)* for more details.  
**Apply Theme...** Opens a file browser dialog for choosing a layout theme file to be applied to the sheet object(s). See *Layout Themes (page 601)* for more details.

In the **Tab Settings** group you may define a background color for the sheet tab.

**Classic Gray** Select this option to use a gray tab coloring with black text.  
**Sheet Background** Sets the tab color to the color of the sheet.  
**Custom Colors** This setting lets you specify a custom **Tab Color** and **Tab Text Color** by clicking the corresponding buttons.  
**Apply to All** Applies the tab color settings to all sheets of the document.



## 31.6 Select Fields/Sheet Properties: Fields



### *Sheet Properties, Fields*

This is where you select *Fields* (page 171) to be displayed on the sheet from the **Available Fields** list. Select the items to be used/removed by clicking them. Use the **Add >** or the **< Remove** button to move them to the desired column.

The selected fields will be displayed in default configured list boxes on the active sheet. To set the specific properties of a list box, right-click on it and select **Properties** from the object float menu (or from **Object** in the **Main** Menu).

To make the system fields selectable, check the **Show System Fields** option.

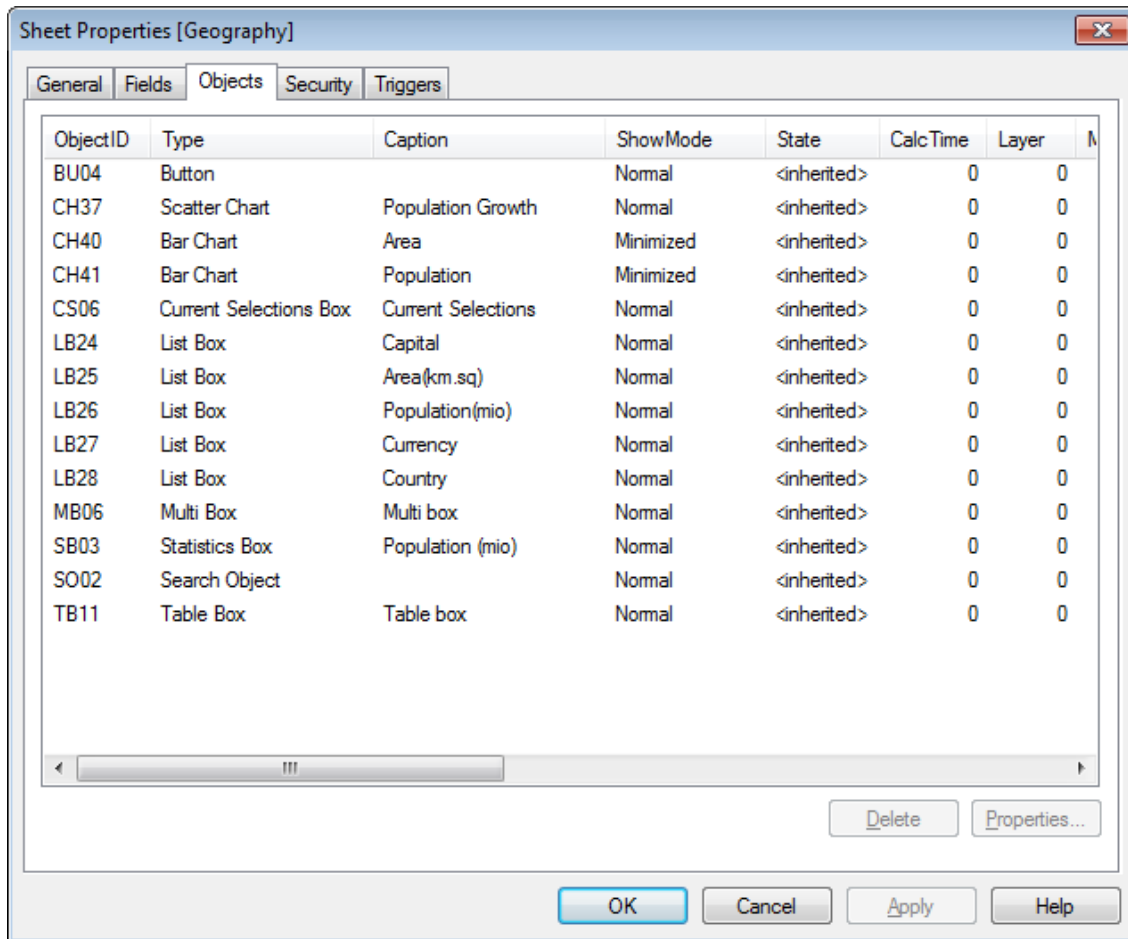
### **Show Fields from Table**

From here, you control what fields appear in the **Available Fields** list. The drop-down list displays the alternative **All Tables** by default.

The alternative **All Tables (Qualified)** shows the fields qualified by the name of the table(s) they occur in. This means that key (connecting) fields will be listed more than once. (This alternative is only used for viewing purposes and has nothing to do with *Qualify* (page 264) fields in the load script.)

It is also possible to view the fields of one table at a time.

## 31.7 Sheet Properties: Objects



*Sheet Properties, Objects*

The **Sheet Properties: Objects** page helps you to keep track of all sheet objects residing on the current sheet. This makes it much easier to get hold of objects that are hidden or have show conditions.

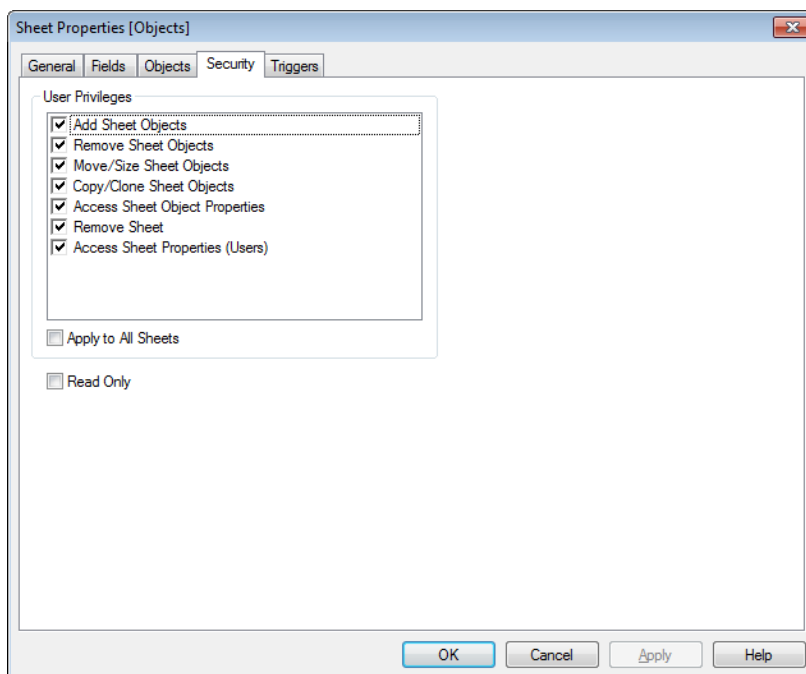
Sort the table on any column, by clicking its column header. The columns contain the following information:

<b>Object ID</b>	Lists the unique <b>Object ID</b> of every existing sheet object. Linked objects have the same object ID.
<b>Type</b>	Clarifies the object type.
<b>Caption</b>	Gives the text, if any, that is displayed in the caption of the object.
<b>ShowMode</b>	Indicates whether the object is normal, minimized, maximized, hidden or has a show condition.
<b>State</b>	Displays which state the object is in. See <i>Alternate States (page 113)</i> for additional information.
<b>CalcTime</b>	The time in milliseconds needed for the last recalculation of the object's contents. This information is only relevant for objects that are actually calculated (charts, table boxes and any objects containing formulas).

<b>Layer</b>	The layers are set on the <b>Sheet Object Properties: Layout</b> page where <b>Bottom</b> , <b>Normal</b> and <b>Top</b> correspond to the numbers -1, 0 and 1. Choose <b>Custom</b> to enter a value of your choice. Valid layer numbers are -128 to 127.
<b>Memory</b>	The amount of transient memory needed for the last recalculation of the object's contents. This information is only relevant for objects that are actually calculated (charts, table boxes and any objects containing formulas).
<b>Left, Top, Right, Bottom</b>	The current position of the indicated edge of the sheet object relative to the edge of the document window, expressed in pixels.
<b>Width, Height</b>	The size dimensions of the object, expressed in pixels.
<b>Left (Min), Top (Min), Right (Min), Bottom (Min)</b>	The current position of the object when iconized (=minimized).
<b>Width (Min), Height (Min)</b>	The dimensions of the object's icon.

After clicking an object in the list, you can use the buttons to either **Delete** it or open its **Properties** dialog directly from this menu. Multiple objects can be selected with Ctrl+Click.

## 31.8 Sheet Properties: Security



### *Sheet Properties, Security*

In this property page you may set the **User Privileges** at the sheet level. By default, all options are checked (on). Persons with administrator privileges can always access this dialog and change the settings. The settings can however prevent ordinary users from altering the sheet layout in a number of ways. The options are:

<b>Add Sheet Objects</b>	Permission to add new sheet objects.
<b>Remove Sheet Objects</b>	Permission to remove any of the sheet objects.

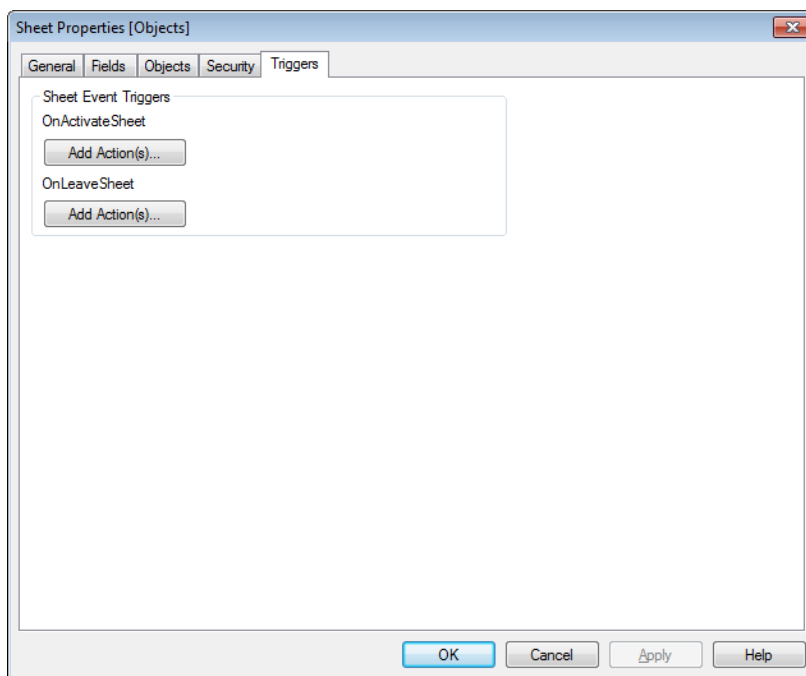
<b>Move/Size Sheet Objects</b>	Permission to change the layout of existing sheet objects.
<b>Copy/Clone Sheet Objects</b>	Permission to add a copy of existing sheet objects.
<b>Access Sheet Object Properties</b>	Permission to access the <b>Sheet Properties</b> dialog.
<b>Remove Sheet</b>	Permission to delete the sheet.
<b>Access Sheet Properties (Users)</b>	Permission to access the different property pages of sheet objects.

Checking the **Apply to All Sheets** option, applies the above options to all sheets of the document.

<b>Read Only</b>	Check this alternative to lock all sheet objects on this sheet for selections. This means that the sheet objects can only display the results of selections made in objects residing on other sheets.
------------------	---

Further privileges on document level can be set in the *Document Properties: Security (page 438)* property page.

## 31.9 Sheet Properties: Triggers



*Sheet Properties, Triggers*

On this page you can specify actions, including macros, to be invoked on sheet events. For more information on macros, see the *Internal Macro Interpreter (page 893)*.

Not all actions can be invoked from the following event triggers. Read more about the different actions under *Add Action Dialog (page 553)*.

---

### Note!

There are limitations as to the behavior of certain macro triggers when working with documents on Qlik-

---

---

View Server. Please study the section *Using Macros in QV Documents on the QV-Server (page 903)* 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 this 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 552)* page. On this page you can add an action to the event. Choose the action **Run 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.

## 31.10 Tabrow Properties

The properties of the tabrow are set in the **Tabrow Properties** dialog, which is opened by right-clicking the tabrow.

**Font**

See *Font (page 491)*. Here you can change the font of the text on the tab. Specify the font, style and size of the text. The font chosen applies to the text on all the tabs.



## 32 Sheet Objects

The following types of sheet objects exist:

<b>List Box</b>	<i>page 473</i>
<b>Statistics Box</b>	<i>page 501</i>
<b>Multi Box</b>	<i>page 507</i>
<b>Table Box</b>	<i>page 521</i>
<b>Charts</b>	<i>page 607</i>
<b>Input Box</b>	<i>page 539</i>
<b>Current Selections Box</b>	<i>page 533</i>
<b>Button</b>	<i>page 549</i>
<b>Text Object</b>	<i>page 561</i>
<b>Line/Arrow Object</b>	<i>page 567</i>
<b>Slider/Calendar Object</b>	<i>page 571</i>
<b>Custom Object</b>	<i>page 595</i>
<b>Search Object</b>	<i>page 583</i>
<b>Bookmark Object</b>	<i>page 579</i>
<b>Container</b>	<i>page 589</i>

### 32.1 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 a 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 *Server Objects Pane* (page 599) 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 *Server Objects Pane* (page 599), accessible from the **View** menu.

### 32.2 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.

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.

If there is more than one active sheet object, the Object menu contains the commands that are common to the active sheet objects.

For a description of the different object menus, see the section on the specific object.

## 32.3 Notes and Comments

Notes and comments can be added to all objects when connected to a QlikView server, allowing users to create and share notes about the current object.

Notes are shared with other QlikView users, and these users can respond by leaving their own comments. It is also possible to save a snapshot (bookmark) of the data with each note.

When right-clicking an object, it is possible to add a new note and to view existing notes, by selecting **Notes** from the context menu. After having selected **Add a New Note** or **View Attached Notes**, all objects in the current document with existing notes are supplied with an indication in the top left corner. The number of attached notes for each object is displayed in the indication.

## 32.4 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 client computer where the Microsoft Office document is used.

---



## 33 List Box

Country	
Argentina	3
Austria	2
Belgium	2
Brazil	9
Canada	3
Denmark	2
Finland	2
France	11

The list box is the most basic sheet object. It contains a list of all possible values of a specific field. Each row in the list box can represent several records in the loaded table, all with identical values. Selecting one value may thus be equivalent to selecting several records in the loaded table.

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.

Right-click on the list box to display the *List Box: Object Menu* (page 473). It can also be accessed from the **Object** menu when the list box is the active object.




### 33.1 List Box: Object Menu

Right-click on a list box and a float menu appears. This menu can also be found under **Object** in the Main menu when a list box is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <b>List Box Properties</b> dialog where <i>List Box Properties: General</i> (page 476), <i>List Box Properties: Expressions</i> (page 479), <i>List Box Properties: Sort</i> (page 486), <i>List Box Properties: Presentation</i> (page 487), <i>Properties: Number</i> (page 490), <i>Font</i> (page 491), <i>Layout</i> (page 492), and <i>Caption</i> (page 496) parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Cycle</b>	Only available when the list box is set to display a <i>Non-Hierarchical Groups (Cyclic)</i> (page 874). Opens a cascade menu with the fields of the group for direct selection of which field to be displayed in the list box.
<b>Search</b>	Opens the <i>Search</i> (page 105) box.
<b>Fuzzy Search</b>	Opens the text search box in <i>Fuzzy Search</i> (page 107) mode.
<b>Advanced Search</b>	Opens the <i>Advanced Search Dialog</i> (page 108). The dialog makes it possible to enter advanced search expressions. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+F.

<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Change Value</b>	<p>Only available for list boxes displaying input fields. Sets the cell clicked on in input edit mode. Equivalent to clicking the input icon in the cell.</p>
<b>Restore Values</b>	<p>Only available for list boxes displaying input fields. Opens a cascade menu with three options:</p> <p><b>Restore Single Value</b> Restores the value in the field value clicked on to its default value from the script.</p> <p><b>Restore Possible Values</b> Restores the values of all possible field values to their default values from the script.</p> <p><b>Restore All Values</b> Restores the values of all field values to their default values from the script.</p>
<b>Select Possible</b>	<p>All non-excluded values in the list box are selected.</p>
<b>Select Excluded</b>	<p>All excluded values in the list box are selected.</p>
<b>Select All</b>	<p>All values in the list box are selected.</p>
<b>Clear</b>	<p>Clears all the current selections in the active list box.</p>
<b>Clear other fields</b>	<p>Clears the selections in all the other sheet objects, while maintaining the ones in the current list box.</p>
<b>Lock</b>	<p>Locks the selected value(s) in the active list box. (Grayed if no selections have been made).</p>
<b>Unlock</b>	<p>Unlocks the locked value(s) in the active list box. (Appears instead of <b>Lock</b> if selections have been locked).</p>
<b>Create Quick Chart...</b>	<p>Opens the <i>Quick Chart Wizard</i> (page 767) that helps you create a chart in a quick and easy way.</p>
<b>Create Time Chart...</b>	<p>Opens the <i>Time Chart Wizard</i> (page 775) that makes it possible to build a chart where a given measure (expression) should be qualified and often compared by different time periods.</p>
<b>Create Statistics Box</b>	<p>Performs various statistical calculations on selected values of a field (sample) and displays the results in a <i>Statistics Box</i> (page 501).</p>

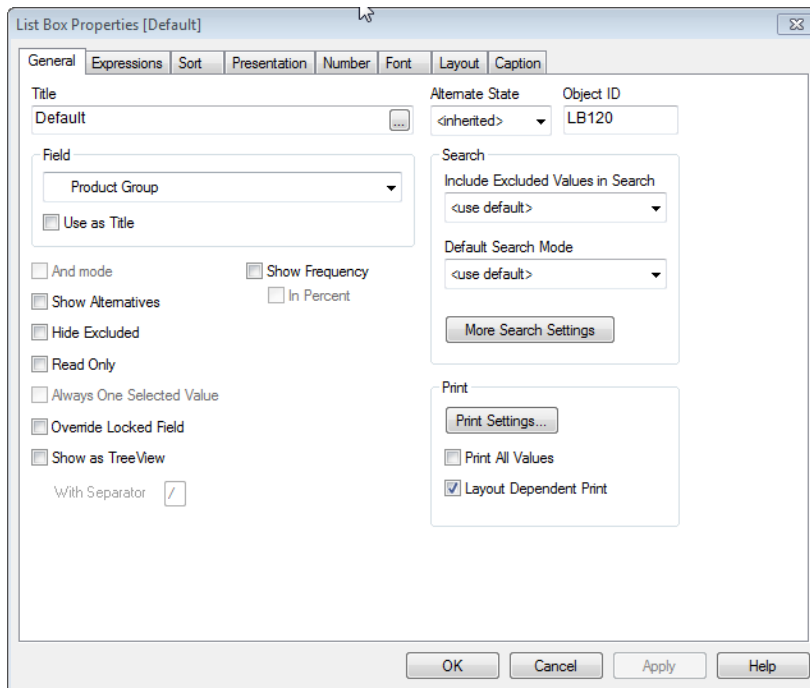
---

<b>Print Possible...</b>	Opens the <i>Print: General (page 91)</i> dialog where print settings can be specified. All non-excluded values of the active list box are sent to the printer.
<b>Print Possible as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView homepage and installed separately.
<b>Send to Excel</b>	Exports possible (selected and optional) values to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. The exported 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.
	<hr/> <b>Note!</b> Mini charts will not be displayed when exporting to Excel! <hr/>
<b>Export...</b>	Opens the <b>Save as</b> dialog where path, file name and (table) file type for the exported data content can be specified.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Possible Values</b> Copies the possible (selected and optional) values to the clipboard. <b>Cell Value</b> Copies the text value of the list box cell right-clicked upon (when invoking the Object menu) to the clipboard. <b>Image</b> 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 <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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 QlikView help.
- Remove** Removes the active sheet object from the sheet.

## 33.2 List Box Properties: General



### *List Box Properties, General*

The **List box Properties: General** tab is opened by a right-click on a list box and choosing the **Properties** command in the float menu. Here it is possible to set general parameters of the list box.

- Title** The text to appear in the list box caption. The default text is the name of the selected field. The title can also be defined as a *Calculated Formula* (page 871) for dynamic update of the label text. Click the **...** button to open the **Edit Expression** dialog for easier editing of long formulas.

**Field**

Select a field from the list. The list box will contain the values of the selected field.

A list box may instead display a cyclic or drill-down group. Only one field will be shown at a time, though. Such groups are created from the *Document Properties: Groups* (page 442) dialog and will be listed with the available fields.

If a *Hierarchic Groups (Drill-Down)* (page 873) 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 of the group. Drilling back up can be accomplished by clicking the drill-up icon appearing in the list box caption.

If a *Non-Hierarchic Groups (Cyclic)* (page 874) is assigned to a list box it can be made to switch to show the next field in the group by clicking on the cycle icon appearing in the list box caption. It is also possible to go directly to any field in the group by using the **Cycle->** option on the list box's object menu.

Furthermore it is possible to display a calculated field in a list box. A calculated field does not come out of script execution, but is defined from an expression. The same rules applies on **Calculated Dimensions** for more information see *Add calculated dimension...* (page 622) . If a calculated field should be used in the list box, select <Expression> in the drop-down after which the **Edit Expression** dialog will open where it is possible to define the expression describing 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 a field name. The button opens the **Edit Expression** dialog where you can define the expression describing the calculated field.

**Note!**

When a calculated field is used, some list box functionality will be different from 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.

**Note!**

It is not possible to use MEASURE fields defined with the *Direct Discovery* (page 161) function in a listbox.

**And mode**

A *Multiple Selection Within a Field* (page 101) must be of either **AND** or **OR** type. By default, the selection is of **OR** type. This command toggles the logical mode by which multiple value selections are made. This menu option is disabled if the field does not satisfy certain conditions, see *AND-Mode in List Boxes* (page 110). This option is not available for calculated fields.

<b>Show Alternatives</b>	Toggles whether a selection directly in the active field list box is to exclude the other values of the field or not. If <b>Show Alternatives</b> is checked, all values except the selected value will be displayed as Optional but may become excluded by association with selections in other fields. Otherwise, all other values are shown as excluded by default. This option is not available for calculated fields.
<b>Hide Excluded</b>	Toggles whether the excluded values of the field should be shown or not. Excluded values become unselectable as well.
<b>Read Only</b>	Checking this option will prevent you from making selections in the list box. However, it will still reflect selections made elsewhere in the document.
<b>Always One Selected Value</b>	Checking this option assures that only one value can be selected, which is useful in documents where the data only makes sense when one value is already selected, such as one currency rate. This option is enabled after one value is already selected in the list box. This option is not available for calculated fields.
<b>Override Locked Field</b>	Checking this option enables selections to be made in the list box even if it is locked. The list box will still be locked for selections made elsewhere in the document.
<b>Show as TreeView</b>	Shows the list box as a <i>List Box Tree View (page 498)</i> . 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 <b>Path</b> parameter of the <b>Hierarchy</b> prefix.
<b>With Separator</b>	Sets the character that should be interpreted as separator in the path used for the <b>Tree View</b> .
<b>Show Frequency</b>	Toggles the status for whether the frequency of a field value is shown or not. By frequency is meant the number of selectable combinations in which the value occurs. This option is not available for calculated fields.
<b>In Percent</b>	Toggles the status for whether the frequency should be shown in absolute numbers or as percentages of the total number of entries.
<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .
<b>Object ID</b>	This is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for list boxes start with LB01. Linked objects share the same object ID. You may edit this ID number later on.
<b>Include Excluded Values in Search</b>	The drop-down list lets you choose between <b>Yes</b> , <b>No</b> or <b>&lt;use default&gt;</b> which applies the default setting as specified under <b>User Preferences: General</b> .

**Default Search Mode** Specifies the initial default search mode to be used in text searches. The mode can always be changed on the fly by typing \* or ~ as part of the search string. The following alternatives are available:

**Use <default>**

The default as specified under *User Preferences (page 69)* will apply.

**Use Wildcard Search**

The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search.

**Use Fuzzy Search**

The initial search string will be a tilde (~) to denote a fuzzy search.

**Use Normal Search**

No additional characters will be added to the search string. Without wildcards, a Normal Search will be made.

**Use Associative Search**

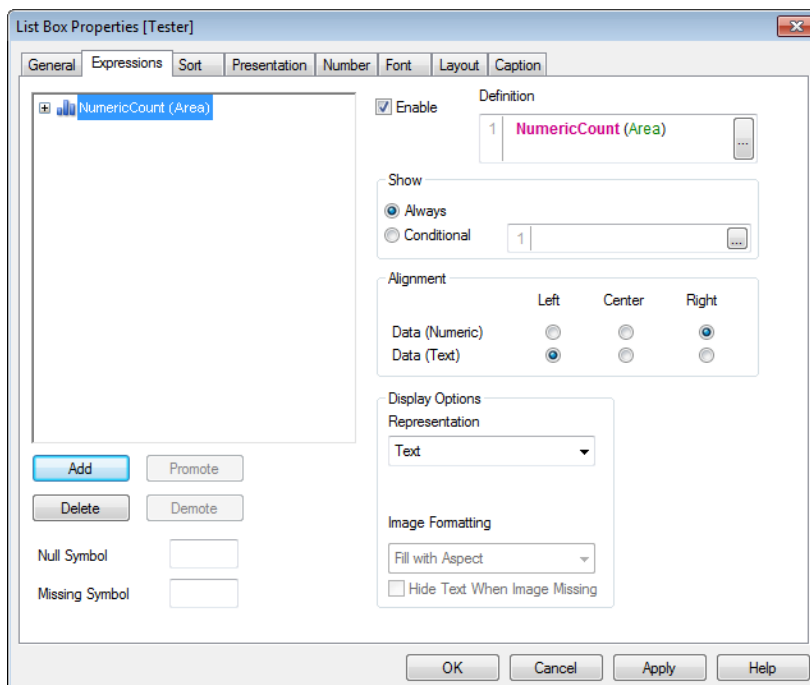
Associative search will be used in the search box. For more information about associative search, see *Search (page 105)*.

**Print Settings** Here you set the print settings for the print job.

**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 values) 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 etcetera. This may be useful e.g. when including list boxes in reports.

## 33.3 List Box Properties: Expressions



*List Box Properties, Expressions*

On the **Expressions** tab you define expressions to be displayed in the list box. Each expression will be placed in a new column in the list box. The upper left corner of the dialog contains a list of the expressions.

## The Expression List

The expression list in the top-left pane is a complete tree control with numerous control options. In front of each expression (or expression group) an expansion icon (a box with a '+' ) is shown. Clicking the icon opens up underlying sub-expressions or attribute expressions. The icon is simultaneously replaced by a collapse icon ('-').

Expression data can also 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 supercede default formatting of the data plots in terms of color etc.

- |                         |  |
|-------------------------|--|
| <b>Background color</b> | Click on <b>Background Color</b> 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 <i>Color Functions</i> (page 375). If the result of the expression is not a valid color representation, black will be used.          |
| <b>Text Color</b>       | Click on <b>Text Color</b> 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 <i>Color Functions</i> (page 375). If the result of the expression is not a valid color representation, black will be used. |
| <b>Text Format</b>      | Click on <b>Text Format</b> 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.   |
| <b>Pie Popout</b>       | Only applicable on pie charts. Click on <b>Pie Popout</b> 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.  |
| <b>Bar Offset</b>       | Only applicable on bar charts. Click on <b>Bar Offset</b> 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.

Several other settings are possible.

**Add** By clicking this button, you get to the *Edit Expression Dialog (page 757)* dialog, 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 object or into any other QlikView object 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 object.

**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 as** dialog from which you can choose the destination of the export file. The file will receive an extension, for example 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.

<b>Import...</b>	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.
<b>Promote</b>	Promotes the selected expression one step up in the list. Use this button if you want to change the order of the expressions.
<b>Demote</b>	Demotes the selected expression one step down in the list. Use this button if you want to change the order of the expressions.
<b>Null Symbol</b>	The symbol entered here will be used for displaying NULL values in the table.
<b>Missing Symbol</b>	The symbol entered here will be used for displaying missing values in the table.
<b>Definition</b>	Shows the composition of the selected expression. It is possible to edit the expression directly in this box. By clicking the ... button the full <i>Edit Expression Dialog (page 757)</i> dialog is opened.
<b>Enable</b>	Marking this check box will enable the selected expression. If the box is not marked, the expression will not be used at all.
<b>Show</b>	If <b>Always</b> is selected, the expression will always be shown in an extra column in the list box. If <b>Conditional</b> is selected, the expression will be dynamically shown or hidden depending on the value of a condition expression entered in the edit box to the right.
<b>Alignment</b>	In this group, the default alignment of your expressions can be set. Text and numbers can be individually set to left-adjusted, centered or right-adjusted.
<b>Display Options</b>	This group defines what will be written in the list box 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.

**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).

**Note!**

The mini chart will not be displayed when exporting to Excel.

**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 *Chart Properties: Presentation (Gauge Chart)* (page 722) for gauge charts.

**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, *Mini Chart Settings* (page 484).

**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.

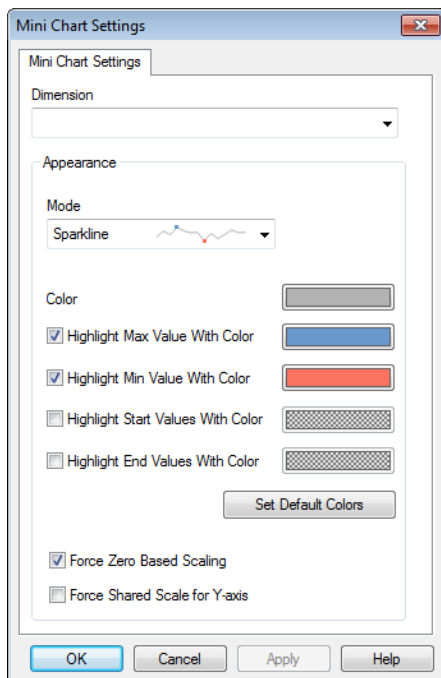
#### Fill with Aspect

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.

## Mini Chart Settings



*The Mini Chart settings dialog*

---

Only available when the **Mini Charts** option has been selected in the **Expressions** page of a straight table's **Properties** dialog. Opens a dialog for setting the properties of the mini chart.

**Dimension** Choose the dimension for which the expression should be plotted.

## Appearance

**Mode** Set the mini chart as Sparklines, line with Dots, Dots, Bars or Whiskers.

**Color** Opens the *Color Area (page 428)* dialog where the 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.

**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** and **Whiskers**.

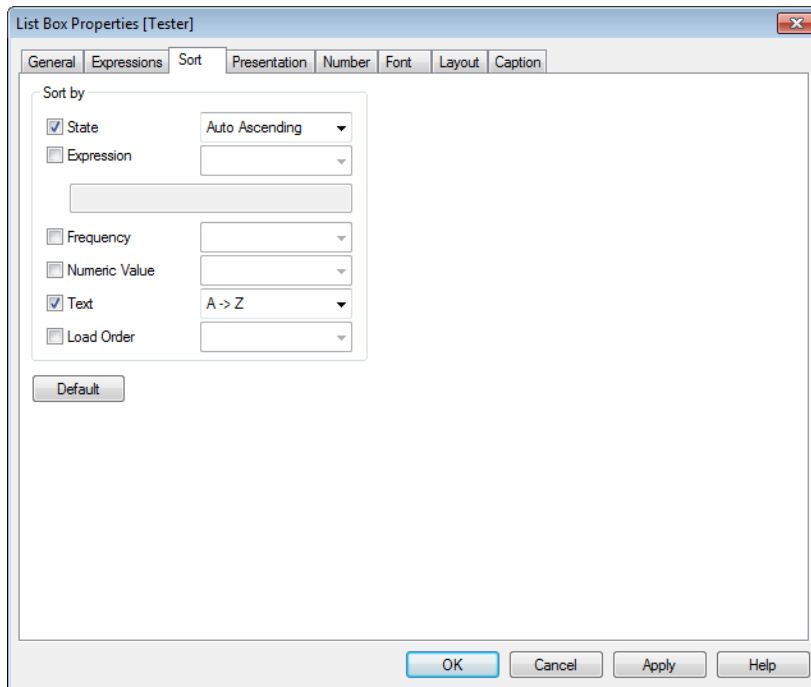
**Highlight End Values With Color** Mark the check box and click on the colored button to open the **Color Area** dialog where the color for the end values can be set. Not available for **Bars** and **Whiskers**.

**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.

## 33.4 List Box Properties: Sort



### *List Box Properties, Sort*

You reach the **Sort** tab by right-clicking a sheet object (List Box, Multi Box, Table Box, Chart or Slider/Calendar object) and choosing **Properties** from the float menu (or **Object, Properties** in the main menu). Here you set the sort order of the values in the sheet object. Some sort options may not be available for some sheet objects.

#### **Sort by:**

##### **State**

Sorts the field values according to their logical state (selected, optional or excluded).

The **Auto Ascending** setting sorts the list box (or, in a multi box, the drop-down list containing the field values) according to **State** only if the list box contains a vertical scroll bar. However, if you enlarge the list box to show all values, the sort order **State** is completely switched off.

##### **Expression**

Sorts the field values according to the expression entered into the text edit box below this sort option. If using sort by expression, an alternative set of records must be defined by a set expression, as the expression is calculated on selected/possible values in the list and the sort only would apply to selected/possible values in the list.

---

**Note!** Sort by expression only works with numeric fields, and not with text fields.

---

See also *Set Analysis (page 819)*.

##### **Frequency**

Sorts the field values by frequency (number of occurrences in the table).

##### **Numeric Value**

Sorts the field values by numeric value.

##### **Text**

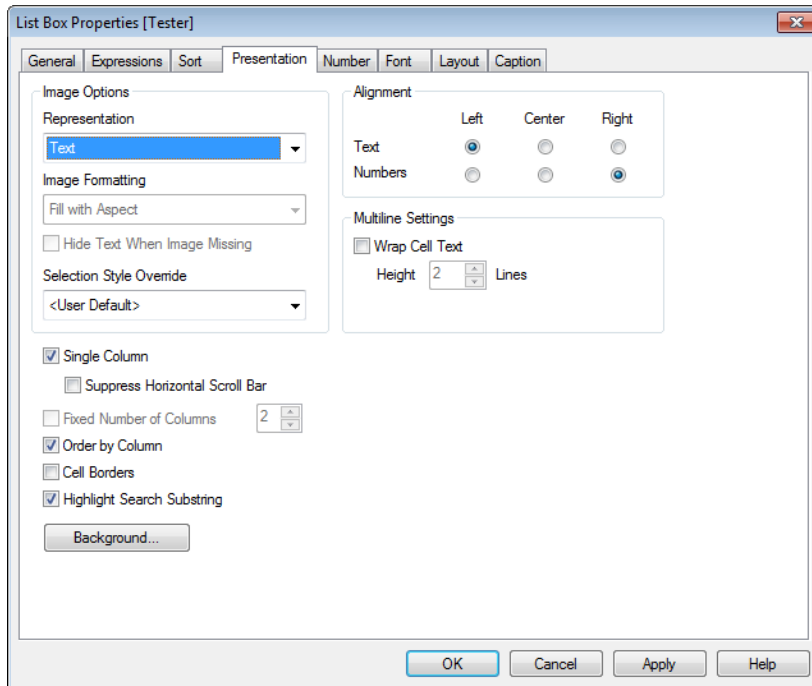
Sorts the field values in alphabetical order.

**Load Order** Sorts the field values by the initial load order.

The **Default** button sort order sets the default sort order.

The order of priority is **State, Expression, Frequency, Numeric Value, Text, Load Order**. Each of these sort criteria can be set to **Ascending** or **Descending**.

## 33.5 List Box Properties: Presentation



### *List Box Properties, Presentation*

The **List Box Properties: Presentation** tab is opened by right-clicking on a list box and choosing the **Properties** command in the float menu. Here it is possible to adjust the layout of the list box cells.

In the **Image Options** group you can let a field value comprising a reference to an image in memory or on disk be interpreted as that image:

Under **Representation** you can choose between **Text** (default), **Image** and **Info as Image**.

When you select **Image**, QlikView will attempt 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 itself (e.g. qmem://<Name>/<Peter>). If QlikView cannot interpret a field value as a valid image reference, the value itself will be displayed, unless the **Hide Text When Image Missing** box is checked.

When you select **Info as Image**, QlikView will display image info linked to the field value via **info load/info select** in the script. If no image info is available for a field value, the value itself will be displayed, unless the **Hide Text When Image Missing** box is checked.

Provided that either image option is selected, the settings under **Image Formatting** can be used for fitting the image into the cell.

**No Stretch** The image is displayed as is, which may cause masking problems.

**Fill** The image is stretched in order to fill the cell, without regard to aspect ratio.

**Keep Aspect** The image is stretched as far as possible while retaining the correct aspect ratio.

**Fill with Aspect** The image is stretched as far as possible while retaining the correct aspect ratio. Areas not covered are then filled by cropping of the image.

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 remains visible. Use the drop-down list under **Selection Style Override** to select a suitable selection style, e.g. **Corner Tag**.

**Single Column** With this option checked, the list box field values will always be presented in a single column.

**Suppress Horizontal Scroll Bar** Check this option to suppress the horizontal scroll bar that is normally displayed when field values are too wide to fit the specified width of the list box. Instead, field values will be truncated as needed.

**Fixed Number of Columns** Check this option to set a fixed number of columns for the list box. The option is unavailable when **Single Column** is selected.

**Order by Column** In list boxes with more than one column, values are displayed row-wise in the sort order specified. The **Order by column** option switches to column-wise display.

**Cell Borders** The field values will be separated by horizontal lines, resembling the rows of a table. **Cell Borders** is automatically activated when the **Wrap Cell Text** option is checked, but may be subsequently deactivated.

**Highlight Search Substring** The search string entered in a search object so far will be highlighted in the matches.

**Background...** Opens the **Background Settings** dialog.

**Alignment** Here you set the alignment of field values in the list box. The alignment of **Text** and **Numbers** is set separately.

**Multiline Settings** In this group the list box cells may be set to display values in multiple rows, which is useful for long text strings.

**Wrap Cell Text**

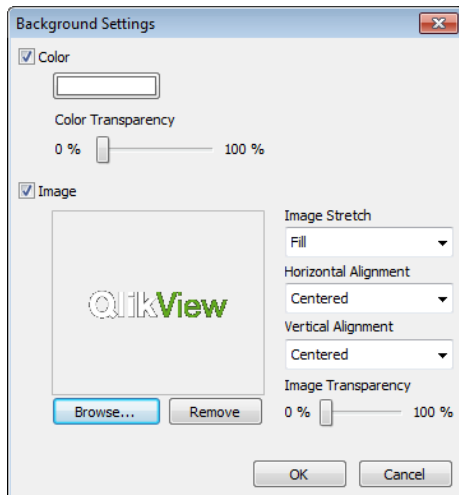
With this option selected, a cell will display its contents in more than one row.

**Height n Lines**

Here the desired limits to the number of cell rows are specified.



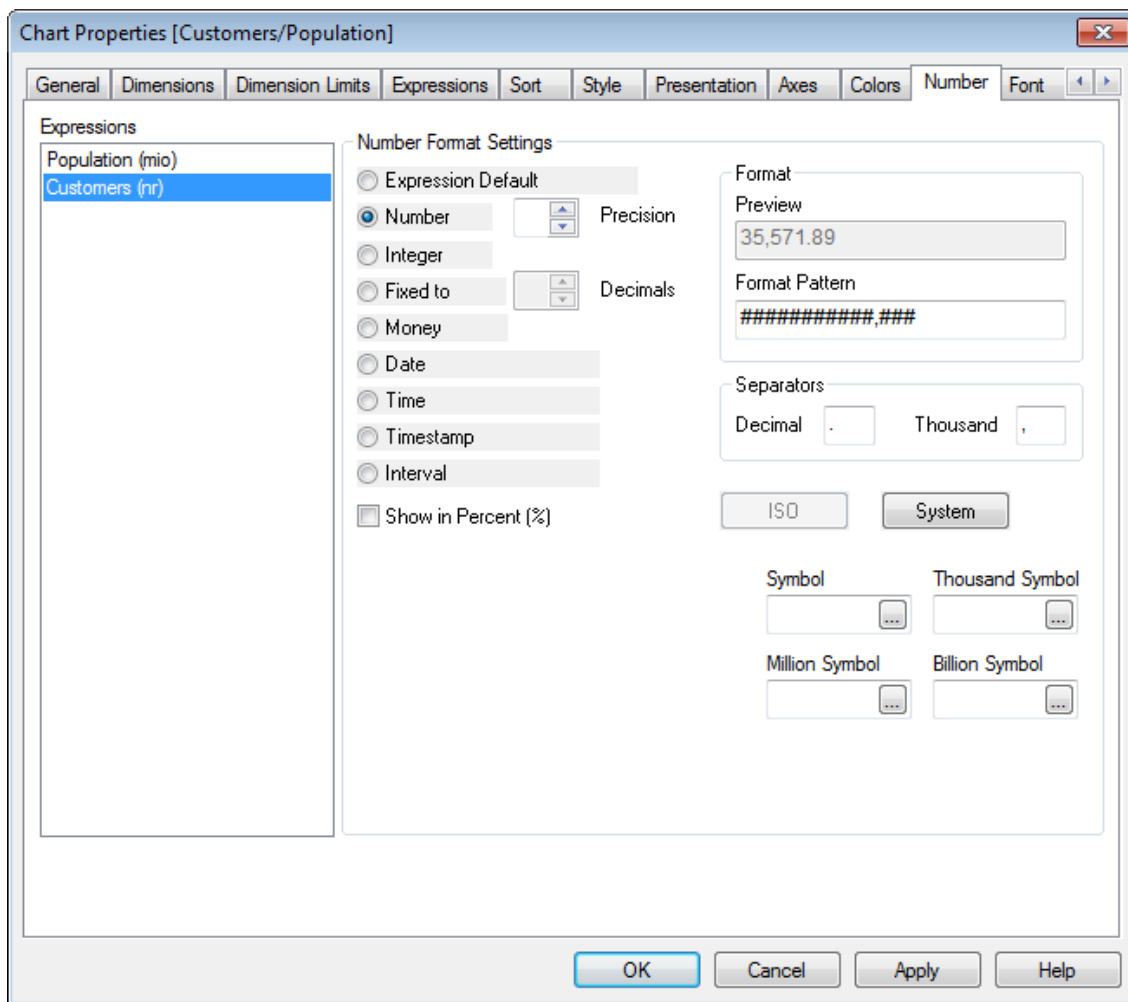
## Background Settings



*The Background Settings dialog*

- Color** If this setting is enabled the background of the object will be colored. Choose a color by clicking on the **Color** button.
- Color Transparency** Only available when the color option has been enabled. Sets the transparency of the background color.
- Image** If this setting is enabled, the background of the object will be an image. Click **Browse** to choose an image. Click **Remove** to remove it from the background.
- Image Stretch** Only available when the image option has been enabled. The setting describes how QlikView formats the image to fit.
- No Stretch**  
The image will be shown as is, without any stretching. This may cause parts of the image to be invisible or only fill part of the background.
- Fill**  
The image will be stretched to fit the background without keeping the aspect ratio of the image.
- Keep Aspect**  
The image will be stretched as far as possible to fill the background while keeping the aspect ratio.
- Fill with Aspect**  
If this option is selected, the image will be stretched to fill the background in both directions while keeping the aspect ratio. This typically results in cropping of the image in one direction.
- Horizontal Alignment** Only available when the image option has been enabled. Aligns the image to the **Left**, **Center** or **Right**.
- Vertical Alignment** Only available when the image option has been enabled. Aligns the image to the **Top**, **Center** or **Bottom**.
- Image Transparency** Only available when the image option has been enabled. Sets the transparency of the background image.

## 33.6 Properties: Number



### *The Number page*

The **Properties: Number** page is opened by right-clicking a sheet object and choosing the **Properties** command from the float menu.

Each field has a default number format which can be set in the *Document Properties: Number (page 449)* page. It is however possible to use a separate number format for an individual sheet object. To do this, check the alternative **Override Document Settings** and specify a number format in the group control below. This property page applies to the active object and contains the following controls for formatting values:

<b>Mixed</b>	Both numbers and text. Numbers shown in original format.
<b>Number</b>	Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.
<b>Integer</b>	Shows numeric values as integers.
<b>Fixed to</b>	Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.

<b>Money</b>	Shows values in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.
<b>Date</b>	Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Time</b>	Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Timestamp</b>	Shows values that can be interpreted as date + time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Interval</b>	Shows time as sequential time increment (e.g. format = <i>mm</i> shows the value as the number of minutes since calendar start ( 1899:12:30:24:00).
<b>Show in Percent (%)</b>	This check box affects the following formats: <b>Number</b> , <b>Integer</b> and <b>Fixed</b> .

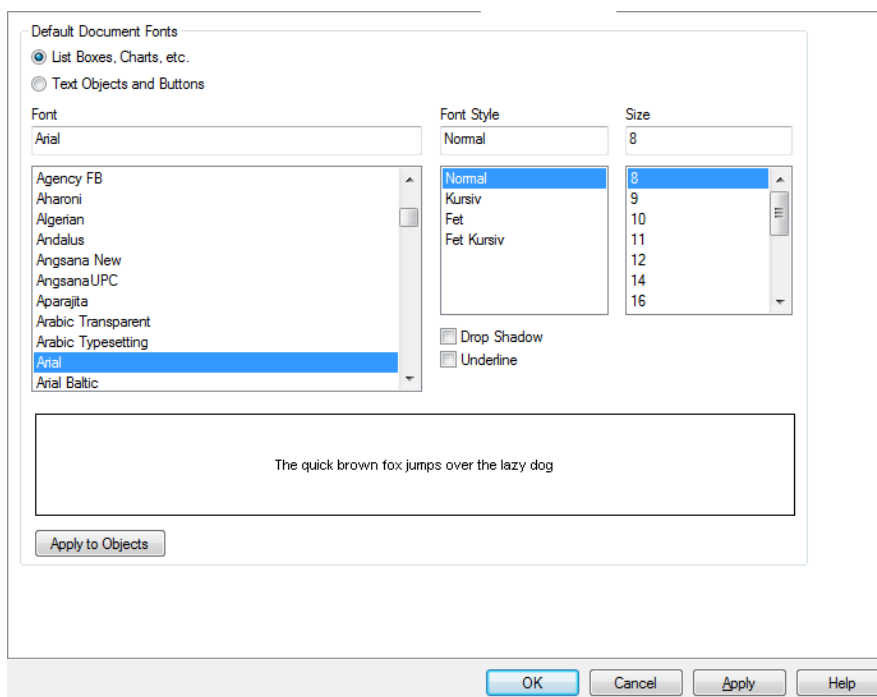
**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

The **ISO** button uses the **ISO** standard for the formatting of date, time and timestamp.

The **System** button applies the settings of the system to the formatting.

The **Change Document Format** button opens the *Document Properties: Number (page 449)* page where the default number format of any field can be edited.

## 33.7 Font



*The Font dialog*

Here the **Font**, **Font style** and **Size** of the font to be used can be set.

The font can be set for any single object (**Object Properties: Font**), or all objects in a document (**Apply to Objects** on **Document Properties: Font**).

Further, the default document fonts for new objects can be set on **Document Properties: Font**. There are two default fonts:

1. The first default font (**List Boxes, Charts, etc**) is used for most objects, including list boxes and charts.
2. The second default font (**Text Objects and Buttons**) is used for buttons and text boxes, which are objects that usually need a larger font.

Finally, the default fonts for new documents can be set on **User Preferences: Font**.

For charts, buttons and text objects (except search objects) a font **Color** can also be specified. The color may **Fixed** (Click the colored button to specify a different color) or it can be dynamically **Calculated** from an expression. The expression must be a valid color representation, which is created using the *Color Functions* (page 375). If the result of the expression is not a valid color representation, the font color will default to black.

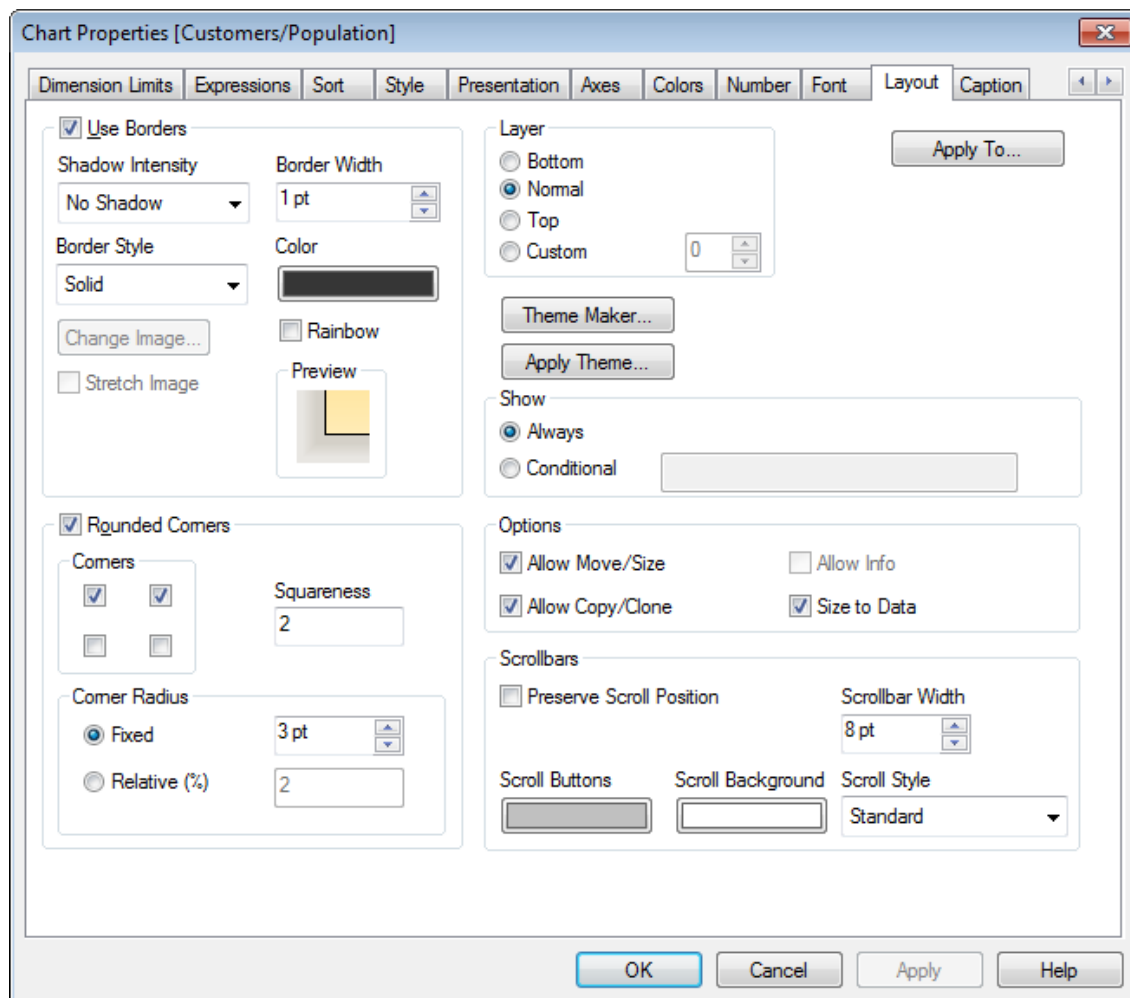
Additional settings are:

**Drop Shadow**                      If this option is checked a drop shadow will be added to the text.

**Underline**                         If this option is checked the text will be underlined.

A sample of the selected font is shown in the preview pane.

## 33.8 Layout



*The Layout page*

A Layout setting will apply to the current object only, if it is made from the Object Properties page.

A Layout setting will apply to all objects of the specified type(s) in the document, if it is made from the Document Properties page.

**Use Borders**

Enable this setting in order to use a border around the sheet object. Specify the type of border by selecting it in the drop-down menu.

**Shadow Intensity** 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 Style** The following predefined border types are available:  
**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, 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.

**Border Width** This option is 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).

**Color** Click this button to open a dialog in which an appropriate base color can be chosen 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 in *Document Properties: General (page 424)*, there is no choice of border type, there is only the **Shadow Intensity** drop-down menu and the **Border Width** setting.

**Rounded Corners**

In the **Rounded Corners** group the general shape of the sheet object is defined. These settings allow for drawing of sheet objects ranging from perfectly circular/elliptical via super elliptical to rectangular:

**Rounded Corners** Checking this option makes the alternatives for rounded corner shapes possible.

**Corners** Corners for which the check box remains unmarked will be drawn rectangular instead.

**Squareness** A variable number between 2 and 100 where 100 defines a rectangle with perfectly square corners and 2 corresponds to a perfect ellipse (a circle for a 1:1 aspect ratio). A squareness between 2 and 5 is usually optimal for achieving rounded corners.

**Corner Radius** This setting determines the radius of the corners in fixed distance (**Fixed**) or in percent of the total quadrant (**Relative (%)**). This setting lets you control the extent to which the corners will be affected by the underlying general shape set under **Squareness**. The 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 an object can be defined as residing in one of three layers:

**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.

### Show

In the **Show** group it is possible to specify a condition under which the sheet object is displayed:

**Always** The sheet object will always be displayed.

**Conditional** The sheet object will be shown or hidden depending on a *Conditional Functions* (page 340) which will be evaluated continuously depending on e.g. selections etc. The sheet object will only be visible when the condition returns TRUE.

---

### Note!

Users with Admin privileges for the document can override all show conditions with the **Show All Sheets and Sheet Objects** in the *Document Properties: Security* (page 438). This functionality can be toggled by pressing Ctrl+Shift+S.

---

### Options

In the **Options** group, it is possible to disallow moving and resizing of the sheet object. The settings in this group are only relevant if the corresponding check boxes are enabled in **Document Properties: Layout** and **Sheet Properties: Security**.

**Allow Move/Size** If this option has been deselected it will be impossible to move or resize the sheet object.

**Allow Copy/Clone** If this option has been deselected it will be impossible to make a copy of the sheet object.

---

**Allow Info** When the *Info* (page 245) function is in use an info icon will be displayed in the window caption whenever a field value has information associated to it. If you don't want the info icon to be displayed in the caption, you can uncheck this option.

**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

Various controls for changing the scrollbar layout are located in the **Scrollbars** group:

**Preserve Scroll Position** With this setting enabled, QlikView will try to preserve the vertical scroll position of tables and charts with an x-axis scroll bar when a selection is made in another object. The setting must be enabled in *User Preferences: Objects* (page 78) as well.

**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. Either color can be defined as a solid color or a gradient via the *Color Area* (page 428) dialog that opens when you click the appropriate button.

**Scrollbar Width** This control affects both the width and the relative size of the scrollbar symbols.

**Scroll Style** Sets the scroll bar style. Select a style in the drop-down control. The **Classic** scroll bar style corresponds to QlikView 4/5 scroll bars. The **Standard** scroll bar style gives a more modern look. The third style is **Light**, which is a thinner, lighter bar.

**Apply To...** Opens the **Caption and Border Properties** dialog where you can set where to apply the properties that have been set on the **Layout** page.

## 33.9 Caption

### *The Caption page*

A Caption setting will apply to the current object only, if it is made from the **Object Properties** page. A Caption setting will apply to all objects of the specified type(s) in the document, if it is made from the **Document Properties** page.

On the **Caption** page, specify layout options that are completely different from the general layout of the object.

**Show Caption** When this option has been checked a caption will be drawn at the top of the sheet object. List boxes and other "box objects" will have the option switched on by default whereas buttons, text objects and line/arrow objects will not.

**Title Text** In the text box you can enter a title to be shown in the caption of the sheet object. Use the **Font...** button to change the caption font.

Set the colors of the caption in its different states. The settings for **Active Colors** and **Inactive Colors** can be made separately from each other.

Click the **Background Color** or **Text Color** button to open the *Color Area (page 428)* dialog. The **Background Color** can be defined as a **solid** or a **gradient** color on the **Color Area** dialog. **Text Color** can be defined a **Fixed** or a **Calculated** color using *Color Functions (page 375)*.

**Wrap Text** If this option is checked, the caption will be displayed in two or more rows.

**Caption Height (Lines)** Set the number of caption lines in this edit box.

The precise size and position of the QlikView object can be determined and adjusted by the size/position settings for the **Normal** or **Minimized** QlikView sheet object. These settings are measured in pixels:

**X-pos** Sets the horizontal position of the left side of sheet object in relation to the sheet's left edge.



<b>Y-pos</b>	Sets the vertical position of the top side of sheet object in relation to the sheet's upper edge.
<b>Width</b>	Sets the width of the QlikView sheet object.
<b>Height</b>	Sets the height of the QlikView sheet object.

The orientation of the caption label can be altered with the **Caption Alignment** options:

<b>Horizontal</b>	The label can be horizontally aligned: <b>Left</b> , <b>Centered</b> or <b>Right</b> within the caption area.
<b>Vertical</b>	The label can be vertically aligned: <b>Top</b> , <b>Centered</b> or <b>Bottom</b> within the caption area.

### 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.

---

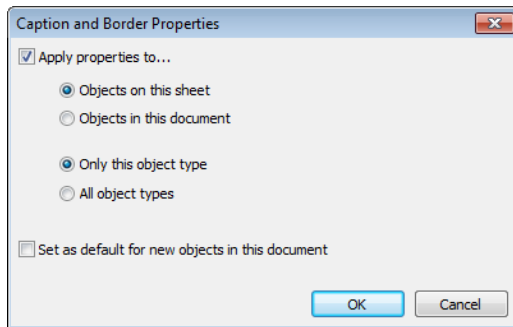
#### Note!

Use the special caption icons with care. Too many icons will only confuse the user.

---

<b>Allow Minimize</b>	When this option has been checked a minimize icon will be displayed in the window caption of the sheet object, providing that the object is minimizable. Furthermore, this will make it possible to minimize the object by double-clicking the caption.
<b>Auto Minimize</b>	This option becomes available when <b>Allow Minimize</b> is checked. When <b>Auto Minimize</b> is checked for several sheet objects on the same sheet, all but one will be automatically minimized at any time. This is useful e.g. for alternately displaying several graphs in the same sheet area.
<b>Allow Maximize</b>	When this option has been checked a maximize icon will be displayed in the window caption of the sheet object, providing that the object is maximizable. Furthermore, this will make it possible to maximizing the object by double-clicking the caption. If both <b>Allow Minimize</b> and <b>Allow Maximize</b> are checked, double-clicking will effect minimizing of the object.
<b>Help Text</b>	Here you may enter a help text to be displayed in a pop-up window. The help text may be specified as a <i>Calculated Formula (page 871)</i> . This option is not available at document level. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas. Enter e.g. a description of the sheet object. A help icon will be added to the window caption of the object. When the mouse pointer is over the icon, the text will be displayed in a pop-up window.

## Caption and Border Properties



*The Caption and Border properties page*

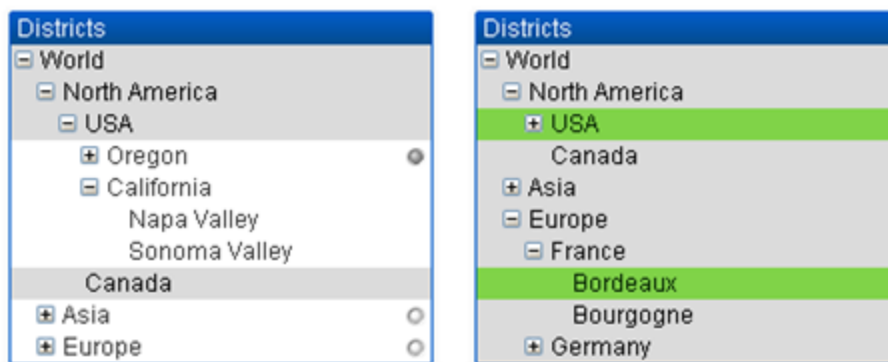
In this dialog you set to which objects in the document the caption and border properties should apply.

- |  |   |
|--|---|
| <b>Apply properties to...</b>                          | <p>Mark this check box to apply the settings to other objects than the current one.</p> <p>1. Choose one of the options:</p> <p><b>Objects on this sheet</b><br/>Applies the settings to objects on the current sheet only. Only available when this dialog is opened from the <b>Layout</b> page of a sheet object.</p> <p><b>Objects in this document</b><br/>Applies the settings to objects of the entire document.</p> <p>2. Choose one of the options:</p> <p><b>Only this object type</b><br/>Applies the settings to all objects of this type. Only available when this dialog is opened from the <b>Layout</b> page of a sheet object.</p> <p><b>All object types</b><br/>Applies the settings to all objects.</p> |
| <b>Set as default for new objects in this document</b> | <p>Mark this check box to use the settings as default for all new objects in the current document. Only available when this dialog is opened from the <b>Layout</b> page of a sheet object.</p>   |

### 33.10 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.

## 33.11 How To Create List Boxes and Table Boxes with Direct Discovery

### List Boxes

List Boxes can be created with DIMENSION fields and MEASURE fields. When using a MEASURE field in a List Box expression, the `aggr()` function should be used to show the aggregated value of the MEASURE field with a DIMENSION field. Selections made with DIMENSION fields in a List Box generate SQL queries on the data source to create the associative table in memory. See *Differences between Direct Discovery and In-Memory Data (page 165)* for an explanation of how Direct Discovery associations are made.

DETAIL fields are not available in List Boxes.

#### Note:

Because List Boxes display only the unique values of a DIMENSION field, the query QlikView generates to display List Box data uses the DISTINCT keyword to retrieve data from the database. However, some databases require that the query instead include a GROUP BY class instead of the DISTINCT keyword. If the Direct Discovery table comes from a database that requires GROUP BY, you use the *Direct-DistinctSupport (page 235)* variable to change the default behavior of the DIMENSION queries.

### Table Boxes

Table Boxes can display all Direct Discovery field types and provide a means to drill to the details of records that contain Direct Discovery fields. Table Boxes are the only QlikView objects in which fields designated DETAIL in the *Direct Query* load statement can be used.

When Direct Discovery fields are used in a Table Box, a threshold is set to limit the number of rows displayed. The default threshold is 1000 records. The default threshold setting can be changed by setting the **DirectTableBoxListThreshold** variable in the load script. For example:

```
SET DirectTableBoxListThreshold=5000
```

The threshold setting applies only to Table Boxes that contain Direct Discovery fields. Table Boxes that contain only in-memory fields are not limited by the **DirectTableBoxListThreshold** setting.

No fields are displayed in the Table Box until the selection has fewer records than the threshold limit.

Direct Discovery fields can be mixed with in-memory fields in a Table Box. Table Boxes that contain only Direct Discovery fields must include a DIMENSION field.

Table Boxes with Direct Discovery data show all selected rows even if they contain identical data. Table Boxes with in-memory data, on the other hand, shows only one of the selected rows when they contain identical data.

## 34 Statistics Box

Population (mio)	
Numeric count	188
Sum	6,825.21
Average	36.30
Min	0.00
Max	1,342.49

The statistics box is a compact way of showing a numeric field in which the separate records are of less interest than e.g. their sum or average. A selection of statistical functions are available. When no specific range of values is selected QlikView treats all the values listed in the corresponding field list box (option values) as the sample.



The default name of the box is the same as that of the field to which the selected values belong.


Right-click on the statistics box to display the *Statistics Box: Object Menu* (page 501). It can also be accessed from the **Object** menu, when the statistics box is the active object.

### 34.1 Statistics Box: Object Menu

By clicking once with the right mouse button when pointing on a statistics box a float menu appears. It contains the following commands:

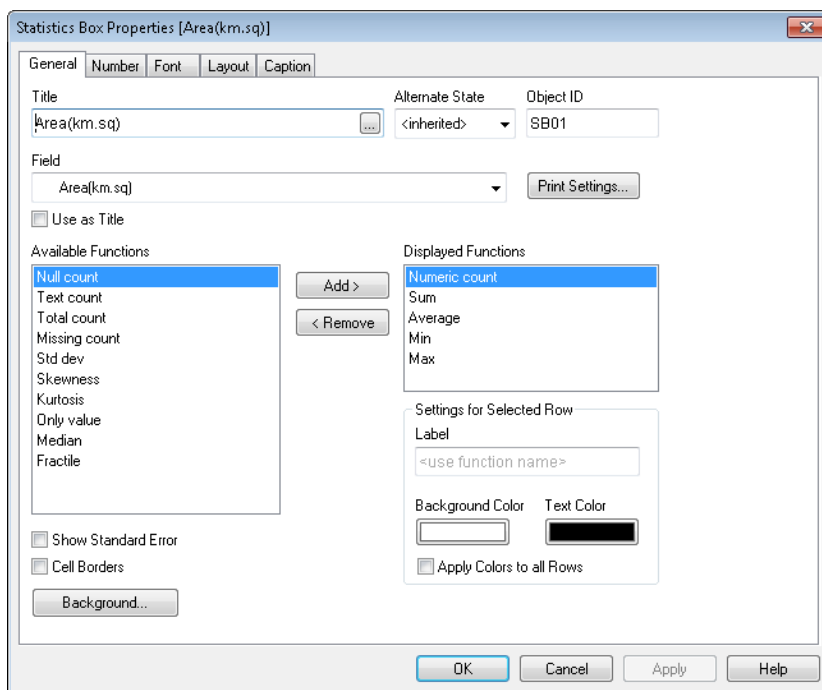
<b>Properties...</b>	Opens the <i>Statistics Box Properties: General</i> (page 503) page. Here it is possible to choose the statistics to be presented in the statistics box. The <i>Statistics Box Properties: Number</i> (page 505), <i>Font</i> (page 491), <i>Layout</i> (page 492), and <i>Caption and Border Properties</i> (page 498) pages can also be reached from this command. Here the font and border parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Order</b>	This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127. <ul style="list-style-type: none"> <li><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</li> <li><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</li> <li><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</li> <li><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</li> </ul>
<b>Select Possible</b>	All non-excluded values in the corresponding list box are selected.
<b>Select Excluded</b>	All excluded values in the corresponding list box are selected.
<b>Select All</b>	All values in the corresponding list box are selected.
<b>Clear</b>	Clears all the current selections in the corresponding list box.

<b>Clear Other Fields</b>	Clears the selections in all the other sheet objects, while maintaining the ones in the corresponding list box.
<b>Lock</b>	Locks the selected value(s) in the corresponding list box.
<b>Unlock</b>	Unlocks the locked value(s) in the corresponding list box.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog to let you print the contents of the statistics box.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send to Excel</b>	Exports possible (including selected) values to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. The exported values will appear in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer.
<b>Export...</b>	Opens the <b>Save as</b> dialog where path, file name and (table) file type for the exported data content can be specified.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Data</b> Copies the data rows of the statistics box to the clipboard. <b>Cell Value</b> Copies the text value of the statistics box cell right-clicked upon (when invoking the Object menu) to the clipboard. <b>Image</b> 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 <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.

- 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 QlikView help.
- Remove** Removes the sheet object.

## 34.2 Statistics Box Properties: General

This property page is opened by selecting **Statistics Box** from **New Sheet Object** in the **View** menu, or by a right-click on a statistics box and choosing **Properties** from the float menu. Here it is possible to choose the statistics to be calculated and displayed in the statistics box.



### *Statistics Box Properties, General*

- Title** Here you can enter a text that will be shown in the sheet object's title area.
- Alternate State** Choose one of the available states in the list. The following Alternate States are always available.
- Inherited**  
The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.
- Default state**  
This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

<b>Object ID</b>	The <b>Object ID</b> is used for <i>Internal Macro Interpreter</i> (page 893) purposes. Every sheet object is assigned a unique ID, which for statistics boxes start with SB01. Linked objects share the same object ID. It is possible to edit this ID number later on.
<b>Field</b>	Select the field that contains the values to be used from the list in the <b>Field</b> group box.
<b>Use as Title</b>	Click the button <b>Use as Title</b> if you want the field name as box title if not, type a title in the <b>Title</b> edit box. The title can also be defined as a <i>Calculated Formula</i> (page 871) for dynamic update of the label text. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas.
<b>Available Functions</b>	Select the statistical functions you want from the list in the <b>Available Functions</b> window and move them to the <b>Displayed Functions</b> window by double-clicking or clicking the <b>Add</b> button.
<b>Show Standard Error</b>	Check <b>Show standard Error</b> to display standard error values for <b>Average</b> and <b>Std dev</b> calculations.
<b>Cell Borders</b>	Check <b>Cell Borders</b> to display separation by horizontal lines, resembling the rows of a table.
<b>Background...</b>	Opens the <i>Background Settings</i> (page 489) dialog.
<b>Displayed Functions</b>	Lists the statistical functions that will be used in the statistics box.
<b>Settings for Selected Row</b>	The appearance of the statistics box can be further modified from the <b>Settings for Selected Row</b> group.
<b>Label</b>	Replace default function names by entering alternative labels here.
<b>Background Color</b>	Specifies the background color of the selected row. The background color can be defined as a solid color or a gradient via the <b>Color Area</b> dialog that opens when you click the button.
<b>Text Color</b>	This alternative lets you pick a text color for the selected row.
<b>Apply Colors to all Rows</b>	Check this alternative before clicking <b>Apply</b> or <b>OK</b> to apply the selected color to all rows of the statistics box.

The available statistical functions are:

<b>Numeric count</b>	Sample size. Number of numeric values among the possible values)
<b>Null count</b>	Number of empty field values among the possible values
<b>Text count</b>	Number of alpha-numeric values among the possible values
<b>Total count</b>	Total number of possible values. This is the same number as the frequency that can be shown in a list box. (sum of <i>Numeric count</i> and <i>Text count</i> )
<b>Missing count</b>	Number of non-numeric values among the possible values (sum of <i>Null count</i> and <i>Text count</i> )
<b>Sum</b>	Sample sum
<b>Average</b>	Sample arithmetic mean (average value)
<b>Std dev</b>	Sample standard deviation

---

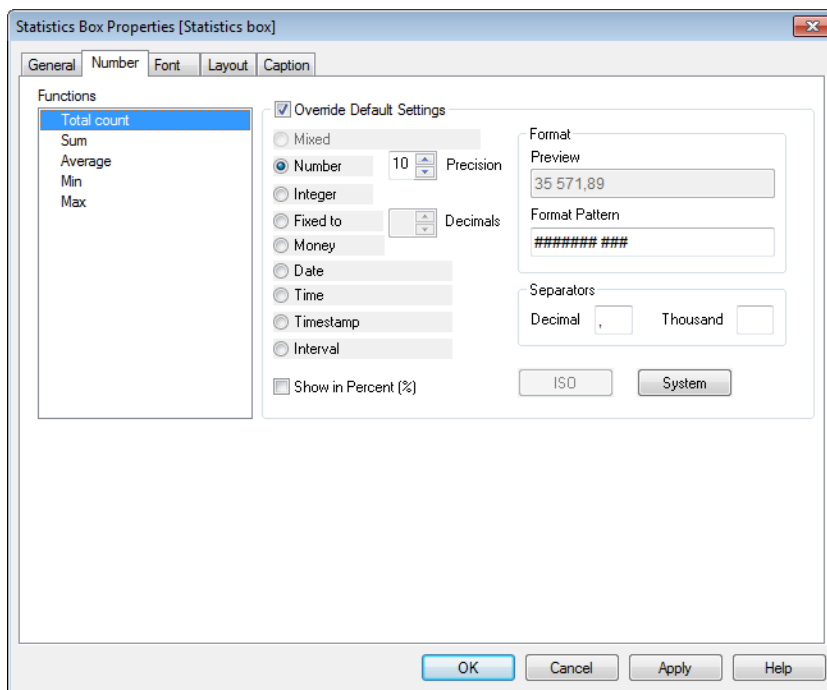


<b>Skewness</b>	Sample skewness
<b>Kurtosis</b>	Sample kurtosis
<b>Min</b>	Sample minimum
<b>Max</b>	Sample maximum
<b>Only value</b>	Only possible numeric value
<b>Median</b>	Sample median
<b>Fractile</b>	Sample fractile

Sample consists of all not excluded (i.e. selected + optional) values of the field.

If the values that represent recognizable field values, such as **Min**, **Max**, are clicked the corresponding field values will become selected.

### 34.3 Statistics Box Properties: Number



#### *Statistics Box Properties, Number*

The **Statistics Box Properties: Number** tab is opened by a right-click on a statistics box and choosing the **Properties** command from the float menu.

Statistically correct number formats for the various statistical functions that can be displayed in statistics box are automatically derived from the base field's data and number format. Here it is possible to set override number formats for individual functions. To do this, select a statistical function from the **Functions** list, check the alternative **Override Document Settings** and specify a number format in the group control below. This property page applies to the active object and contains the following controls for formatting values:

<b>Number</b>	Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.
<b>Integer</b>	Shows numeric values as integers.

<b>Fixed to _ Decimals</b>	Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.
<b>Money</b>	Shows values in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.
<b>Date</b>	Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Time</b>	Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Timestamp</b>	Shows values that can be interpreted as date + time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Interval</b>	Shows time as sequential time increment (e.g. format = <i>mm</i> shows the value as the number of minutes since calendar start ( 1899:12:30:24:00)).

The **Show in Percent (%)** button operates on the following formats: **Number**, **Integer** and **Fixed to**.

**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

The **ISO** button uses the **ISO** standard for the formatting of date, time and timestamp.

The **System** button applies the settings of the system to the formatting.

## 34.4 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 34.5 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 34.6 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 35 Multi Box



Due to its unique option to sort the displayed fields by applicability (This **Sort by Applicability** option is accessed from the *Multi Box Properties: General* (page 510) page), the multi box offers the ultimate solution to the problem of displaying a large number of list boxes on the same sheet.

Right-click on the multi box to display the *Multi Box: Object Menu* (page 507). It can also be accessed from the **Object** menu, when the multi box is the active object.




### 35.1 Multi Box: Object Menu

The multi box **Object** menu has three different versions. If the caption area is clicked the float menu will contain commands that are applicable to the whole box, whereas if an individual field is under the pointer the menu will be extended to include commands that apply to that field. Finally, if you first open a file, the float menu will have yet another layout. The combined commands of the menus are:

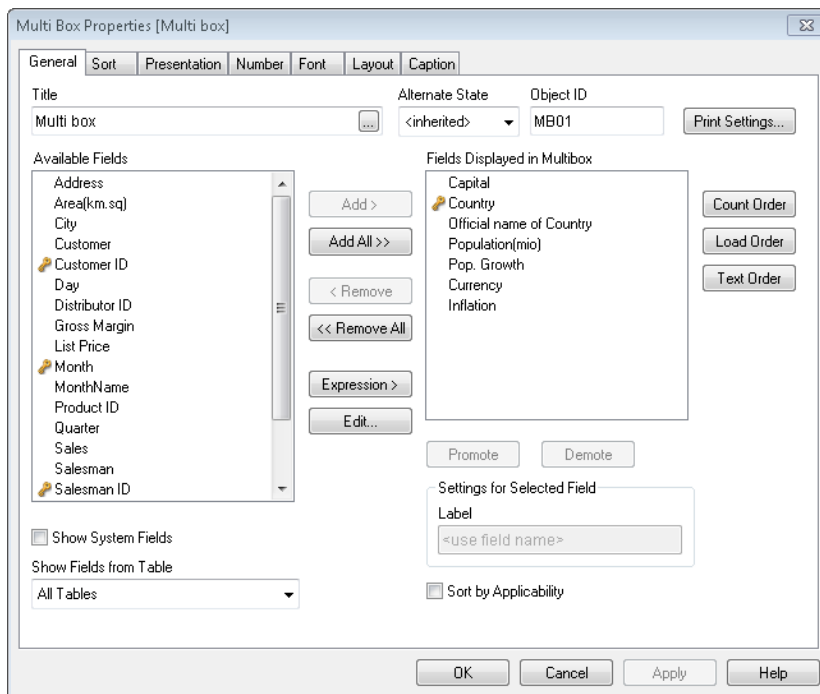
<b>Properties...</b>	Opens the <b>Multi Box Properties</b> dialog where <i>Multi Box Properties: General</i> (page 510), <i>Multi Box Properties: Sort</i> (page 512), <i>Multi Box Properties: Presentation</i> (page 513), <i>Multi Box Properties: Number</i> (page 517), <i>Font</i> (page 491) and <i>Layout</i> (page 492) parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Order</b>	This cascade menu is only available when the <b>Design Grid</b> command of <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127. <ul style="list-style-type: none"> <li><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</li> <li><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</li> <li><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</li> <li><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</li> </ul>
<b>Search</b>	Opens the <i>Text Search Box</i> for search in an opened field.
<b>Fuzzy Search</b>	Opens the text search box in <i>Fuzzy Search</i> (page 107) mode for search in an opened field.

<b>Advanced Search</b>	Opens the <i>Advanced Search Dialog (page 108)</i> for search in an opened field. The dialog makes it possible to enter advanced search expressions. This command can also be invoked by the following keyboard shortcut: Ctrl+Shift+F.
<b>Select Possible</b>	All non-excluded values of the field are selected.
<b>Select Excluded</b>	All <i>Excluded</i> values of the field are selected.
<b>Select All</b>	All values of the field are selected.
<b>Clear All Selections</b>	Clears all the current selections of the multi box.
<b>Clear</b>	Clears all the current selections of the field.
<b>Clear Other Fields</b>	Clears the selections in all the other sheet objects, including selections in other fields of the current multi box, while maintaining selections in this specific field of the current multi box.
<b>Lock</b>	Locks the selected value(s) of the field.
<b>Lock All Selections</b>	Locks the selected value(s) of the multi box.
<b>Unlock</b>	Unlocks the locked value(s) of the field.
<b>Unlock All Selections</b>	Unlocks the locked value(s) of the multi box.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog where print settings can be specified. Printing the multi box is only meaningful when a single value is displayed in each field (either because it's the only selected value or the only associated value).
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send to Excel</b>	Exports the parts of the multi box which are visible when the field lists are closed to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. The exported values will appear in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer. This operation is only meaningful when a single value is displayed in each field (either because it's the only selected value or the only associated value).
<b>Export...</b>	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 <i>QVD Files (page 409)</i> .

---

<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Data</b> Copies the data rows of the multi box to the clipboard.</p> <p><b>Cell Value</b> Copies the text value of the multi box cell right-clicked upon (when invoking the Object menu) to the clipboard.</p> <p><b>Image</b> 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 <b>User Preferences</b> dialog, <b>Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	<p>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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Maximize</b>	<p>Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Restore</b>	<p>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.</p>
<b>Help</b>	<p>Opens QlikView help.</p>
<b>Remove</b>	<p>Removes the sheet object.</p>

## 35.2 Multi Box Properties: General



### Multi Box Properties, General

#### Title

In the **Title** window the multi box can be given a name which will be displayed in the window caption. The title can also be defined as a *Calculated Formula* (page 871) for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

#### Alternate State

Choose one of the available states in the list. The following Alternate States are always available.

##### Inherited

The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.

##### Default state

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

#### Available Fields

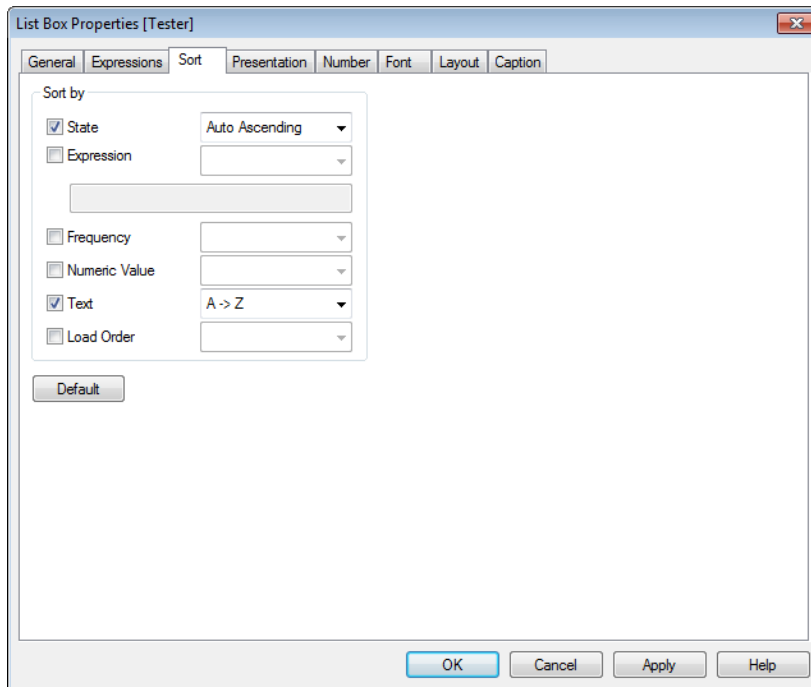
In this column the names of the data source fields are shown. Initially all fields (excluding system fields) appear in this column. To include the *System Fields* (page 379) check **Show System Fields**.

Key fields will be indicated with a key symbol. Select the items to be used/removed by clicking them. Use the **Add >** or the **< Remove** button to move them to the desired column.

---

<b>Fields Displayed in Multibox</b>	<p>In this column the names of the fields selected from the <b>Available Fields</b> list to be included in the multi box are shown. Initially no fields are shown in this column.</p> <p>When a field is selected in this list, you may give it a custom <b>Label</b> in the edit window.</p>
<b>Show Fields from Table</b>	<p>From here, you control what fields appear in the <b>Available Fields</b> list. The drop-down list displays the alternative <b>All Tables</b> by default.</p> <p>The alternative <b>All Tables (Qualified)</b> shows the fields qualified by the name of the table(s) they occur in. This means that key (connecting) fields will be listed more than once. (This alternative is only used for viewing purposes and has nothing to do with <i>Qualify (page 264)</i> fields in the load script.)</p> <p>It is also possible to view the fields of one table at a time.</p>
<b>Expression</b>	<p>Opens the <i>Edit Expression Dialog (page 757)</i> dialog where an expression can be created that can then be used as a displayed field in the multi box.</p>
<b>Edit...</b>	<p>Opens the <i>Edit Expression Dialog (page 757)</i> dialog for the field chosen in the <b>Fields Displayed in Multibox</b> column.</p>
<b>Promote</b>	<p>Moves a field upwards in the display order.</p>
<b>Demote</b>	<p>Moves a field downwards in the display order.</p>
<b>Count Order</b>	<p>Sorts the fields in the <b>Fields Displayed in Multibox</b> column in numerical order.</p>
<b>Load Order</b>	<p>Sorts the fields in the <b>Fields Displayed in Multibox</b> column in load order, i.e. the order in which they are read from the database.</p>
<b>Text Order</b>	<p>Sorts the fields in the <b>Fields Displayed in Multibox</b> column in alphabetical order.</p>
<b>Sort by Applicability</b>	<p>When this option is checked, the sort order of the <b>Fields Displayed in Multibox</b> will be dynamically updated during selections so that fields with non-excluded field values will move upwards in the list while fields without possible values will move downwards. This option will in many cases make it possible to use literally hundreds of fields in a single multi box.</p>
<b>Object ID</b>	<p>This is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for multi boxes start with MB01. Linked sheet objects share the same object ID. You may edit this ID number later on.</p>

## 35.3 Multi Box Properties: Sort



### *List Box Properties, Sort*

You reach the **Sort** tab by right-clicking a sheet object (List Box, Multi Box, Table Box, Chart or Slider/Calendar object) and choosing **Properties** from the float menu (or **Object, Properties** in the main menu). Here you set the sort order of the values in the sheet object. Some sort options may not be available for some sheet objects.

#### **Sort by:**

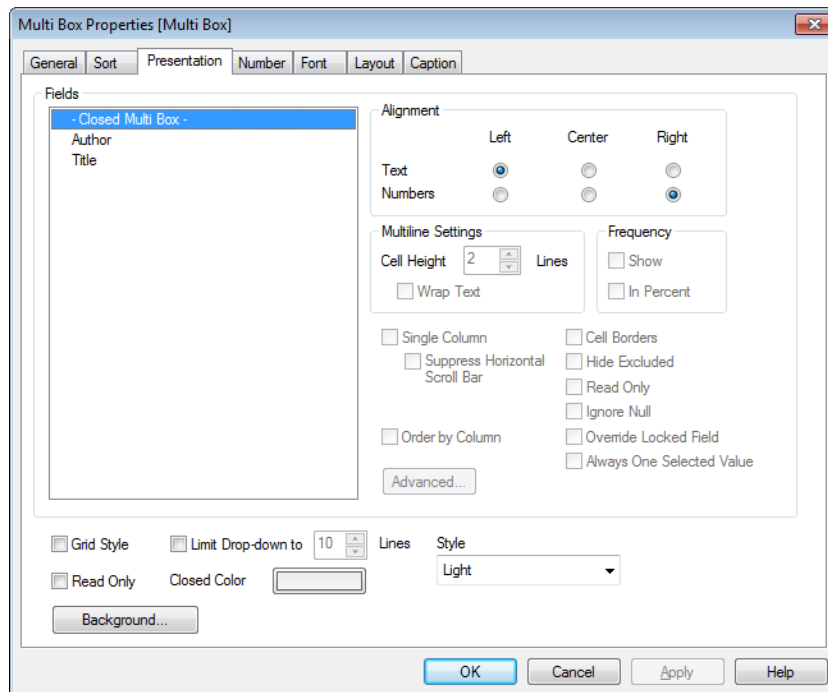
- |                      |  |
|----------------------|--|
| <b>State</b>         | Sorts the field values according to their logical state (selected, optional or excluded).<br>The <b>Auto Ascending</b> setting sorts the list box (or, in a multi box, the drop-down list containing the field values) according to <b>State</b> only if the list box contains a vertical scroll bar. However, if you enlarge the list box to show all values, the sort order <b>State</b> is completely switched off. |
| <b>Expression</b>    | Sorts the field values according to the expression entered into the text edit box below this sort option. If using sort by expression, an alternative set of records must be defined by a set expression. Note that sort by expression here only works with numeric fields and not text fields. See also <i>Set Analysis (page 819)</i> .  |
| <b>Frequency</b>     | Sorts the field values by frequency (number of occurrences in the table).  |
| <b>Numeric Value</b> | Sorts the field values by numeric value.   |
| <b>Text</b>          | Sorts the field values in alphabetical order.  |
| <b>Load Order</b>    | Sorts the field values by the initial load order.  |

The **Default** button sort order sets the default sort order.



The order of priority is **State, Expression, Frequency, Numeric Value, Text, Load Order**. Each of these sort criteria can be set to **Ascending** or **Descending**.

## 35.4 Multi Box Properties: Presentation



### *Multi Box Properties, Presentation*

The **Multi Box Properties: Presentation** tab is opened by a right-click on a multi box and choosing the **Properties** command in the **Object** menu. Here it is possible to adjust the layout of the multi box cells. Adjustments are made independently for the different fields of the multi box. It is also possible to adjust the layout for the - Closed Multi Box -.

#### **Alignment**

Here you set the alignment of field values. The alignment of **Text** and **Numbers** is set separately.

#### **Multiline Settings**

In this group the multi box field cells may be set to display values in multiple rows, which is useful for long text strings.

##### **Wrap Text**

With this option selected, a cell will display its contents in more than one row.

##### **Cell Height/Lines**

Here the desired limit to the number of cell rows is specified.

#### **Frequency**

##### **Show**

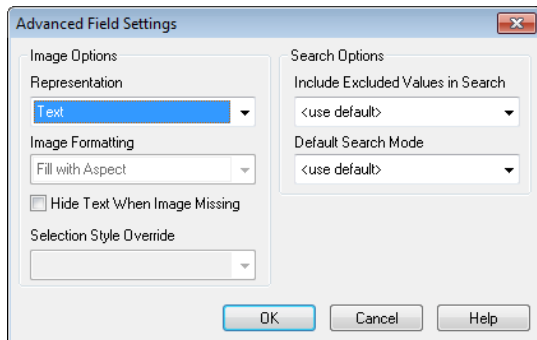
Toggles the status for whether the frequency of the selected field value is shown or not. By frequency is meant the number of selectable combinations in which the value occurs.

##### **In Percent**

Toggles the status for whether the frequency should be shown in absolute numbers or as percentages of the total number of entries.

<b>Advanced</b>	This button opens the <i>Advanced Field Settings (page 515)</i> dialog which offers settings for image representation of field values and special text search options.
<b>Single Column</b>	With this option checked, the multi box field values will always be presented in a single column.
<b>Suppress Horizontal Scroll Bar</b>	Check this option to suppress the horizontal scroll bar that is normally displayed when field values are too wide to for the specified width of the multi box. Instead, field values will be truncated as needed.
<b>Order by Column</b>	In multi box fields with more than one column, values are displayed row-wise in the sort order specified. The <b>Order by column</b> option switches to column-wise display.
<b>Cell Borders</b>	The field values will be separated by horizontal lines, resembling the rows of a table. <b>Cell borders</b> is automatically activated when the <b>Wrap Text</b> option is checked, but may be subsequently deactivated.
<b>Hide Excluded</b>	Toggles whether the excluded field values should be shown or not. Excluded values become unselectable as well.
<b>Read Only</b>	This checkbox disables selections made directly in this particular multi box field, making it a display tool only.
<b>Ignore Null</b>	NULL-values are not considered for the sake of displaying possible values of a specific field.
	<hr/> <b>Note!</b> Incorrect use of this option may lead to the display of non-correlated data in the multi box. <hr/>
<b>Override Locked Field</b>	This check box allows selections to be made in a particular field of the multi box even if that field is locked. The field will remain locked for selections made elsewhere in the document.
<b>Grid Style</b>	This checkbox changes the multi box layout, so that each label is positioned above its corresponding field.
<b>Read Only</b>	This checkbox disables selections made directly in the multi box fields, making it a display tool only.
<b>Closed Color</b>	Sets the color of the data column cells of a closed multi box. A color can be defined as a solid color or a gradient via the <i>Color Area (page 428)</i> dialog that opens when clicking the button.
<b>Limit Drop-down to n Lines</b>	Limits 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.
<b>Style</b>	Choose between the styles <b>Borders</b> and <b>Light</b> .
<b>Background...</b>	Opens the <i>Background Settings (page 489)</i> dialog.

## Advanced Field Settings



### *Advanced Field Settings*

This dialog can be accessed from *Multi Box Properties: Presentation* (page 513), *Table Box Properties: Presentation* (page 528) and *Chart Properties: Dimensions* (page 620). The image options described below are unavailable for bitmap charts.

## Image Options

### 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.

#### 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, unless the **Hide Text When Image Missing** box is checked. The option is not available for bitmap charts.

### 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.

### 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 remains visible. Use the drop-down list under **Selection Style Override** to select a suitable selection style, e.g. **Corner Tag**.

## 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

This setting specifies whether excluded values should be included in text searches. The following alternatives are available:

#### <use default>

The default as specified under *User Preferences (page 69)* applies.

#### Yes

Excluded values are always included in text search.

#### No

Excluded values are never included in text search, whereas optional values are included as usual.

### Default Search Mode

This setting specifies the initial default search mode to be used in text searches. The mode can always be changed on the fly by typing \* or ~ as part of the search string. The following alternatives are available:

#### <use default>

The default as specified under *User Preferences (page 69)* applies.

#### Use Wildcard Search

The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search.

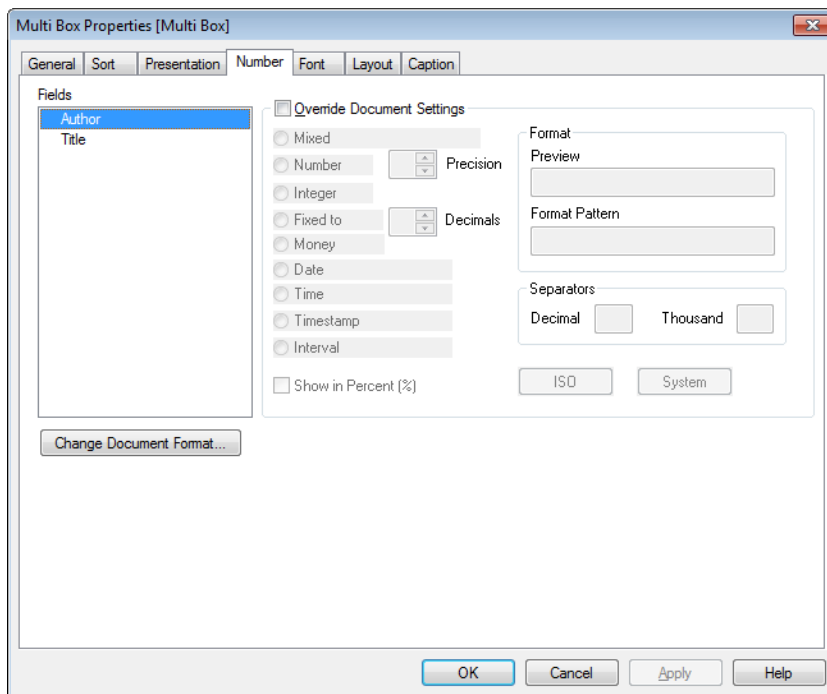
#### Use Fuzzy Search

The initial search string will be a tilde (~) to denote a fuzzy search.

#### Use Normal Search

No additional characters will be added to the search string. Without wildcards, a Normal Search will be made.

## 35.5 Multi Box Properties: Number



*Multi box properties, Number*

The **Multi Box Properties: Number** tab is opened by a right-click on a multi box and choosing the **Properties** command from the float menu.

This property page provides formatting settings for all fields of the multi box. The number formatting can be set individually for fields by selecting one or several fields (Click, Shift-click or Ctrl-click) in the **Fields** text box.

Each field has a default number format which can be set in the *Document Properties: Number (page 449)* page. It is however possible to use a separate number format for an individual sheet object. To do this, check the alternative **Override Document Settings** and specify a number format in the group control below. This property page applies to the active object and contains the following controls for formatting values:

<b>Mixed</b>	Both numbers and text. Numbers shown in original format.
<b>Number</b>	Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.
<b>Integer</b>	Shows numeric values as integers.
<b>Fixed to _ Decimals</b>	Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.
<b>Money</b>	Shows values in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.
<b>Date</b>	Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Time</b>	Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Timestamp</b>	Shows values that can be interpreted as date + time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Interval</b>	Shows time as sequential time increment (e.g. format = <i>mm</i> shows the value as the number of minutes since calendar start ( 1899:12:30:24:00)).

The **Show in Percent (%)** button operates on the following formats: **Number**, **Integer** and **Fixed to**.

**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

The **ISO** button uses the **ISO** standard for the formatting of date, time and timestamp.

The **System** button applies the settings of the system to the formatting.

The **Change Document Format** button opens the *Document Properties: Number (page 449)* page where the default number format of any field can be edited.

## 35.6 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 35.7 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 35.8 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.





## 36 Table Box



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

The table box is a sheet object that shows several fields simultaneously. The content of every row is logically connected. The columns may be fetched from different internal tables, letting the user create tables from any possible combination of fields.

Right-click on the table box to display the *Table Box: Object Menu* (page 521). It can also be accessed from the **Object** menu, when the table box is the active object.

### 36.1 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. Double-click the caption of the column 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. Disable this feature by deselecting the **Allow Drag and Drop** check box in the **Table Box Properties: Presentation** page.




### 36.2 Table Box: Object Menu

The table box **Object** menu has two different versions. If the caption area is clicked the float menu will contain commands that are applicable to the whole box, if an individual field is under the pointer the menu will be extended to include commands that apply to that field. The combined commands of the menus are:

<b>Properties...</b>	Opens the <b>Table Box Properties</b> dialog where <i>Table Box Properties: General</i> (page 525), <i>Table Box Properties: Sort</i> (page 527), <i>Table Box Properties: Presentation</i> (page 528), <i>Chart Properties: Style</i> (page 755), <i>Table Box Properties: Number</i> (page 531), <i>Font</i> (page 491), and <i>Layout</i> (page 492) parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Fit Columns to Data</b>	Adjusts the width of all columns in the tables to the widest data in each column. The header is included in the calculation.
<b>Equal Column Width</b>	Sets the column width in the table equal to that of the column on which you clicked.

<b>Sort</b>	Sorts the records by the field on which you click.
<b>Custom Format Cell</b>	Opens the <i>Custom Format Cell Dialog (page 524)</i> which lets you format cells in the column and stripe which you clicked upon. This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Change Value</b>	Only available for table box columns displaying input fields. Sets the cell clicked on to input mode. Equivalent to clicking the input icon in the cell.
<b>Restore Values</b>	<p>Only available for table box columns displaying input fields. Opens a cascade menu with three options.</p> <p><b>Restore Single Value</b> Restores the value in the field value clicked on to its default value from the script.</p> <p><b>Restore Possible Values</b> Restores the values of all possible field values to their default values from the script.</p> <p><b>Restore All Values</b> Restores the values of all field values to their default values from the script.</p>
<b>Select Possible</b>	All non- <b>excluded</b> values of the field are selected.
<b>Select Excluded</b>	All excluded values of the field are selected.
<b>Select All</b>	All values of the field are selected.
<b>Clear All Selections</b>	Clears selections in all fields displayed in the table box.
<b>Clear</b>	Clears all the current selections of the field.
<b>Clear Other Fields</b>	Selects all possible values in the current field, then clears the selections in all other fields.
<b>Lock</b>	Locks the selected value(s) of the field.
<b>Unlock</b>	Unlocks the locked value(s) of the field.
<b>Print...</b>	Opens the standard <i>Print: General (page 91)</i> dialog, allowing you to print the table.

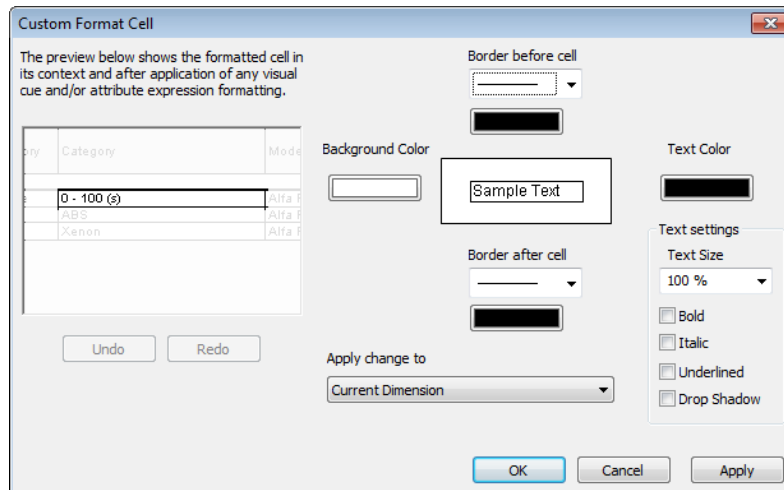
---

<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After hitting <b>OK</b> you will be prompted for a file name for the PDF output file. This command is only available if a QlikViewPDF printer is available on the system. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send to Excel</b>	Exports the table to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. 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.
<b>Export...</b>	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). Images in a table box will not be included when exporting to HTML format.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Full Table</b> Copies the table to the clipboard, complete with header and selection status. <b>Table Data Area</b> Copies only the values of the table to the clipboard. <b>Cell Value</b> Copies the text value of the list box cell right-clicked upon (when invoking the Object menu) to the clipboard. <b>Image</b> 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 <b>User Preferences: Export</b> page. Tables generated by the hidden part of the script will not be represented by name in the <i>\$Table</i> system field. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.

---

**Remove** Removes the active sheet object from the sheet.

## Custom Format Cell Dialog



### *Custom Format Cell*

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** command of *View Menu* (page 52) is activated or the *Always Show Design Menu Items* (page 78) checkbox is marked. 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 striping 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. 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.

**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.

**Sample Text** This is the preview cell which shows the settings made.

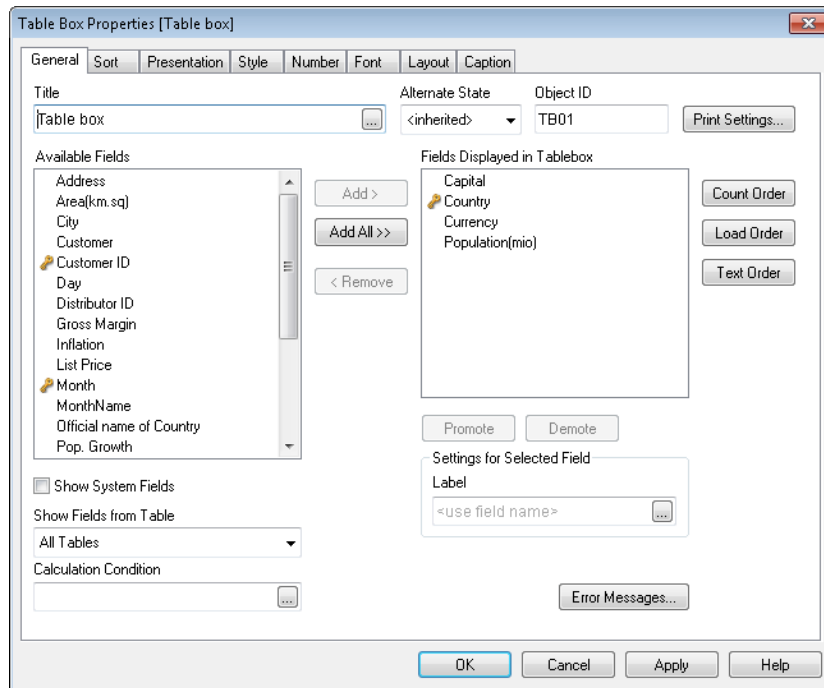
---

<b>Background Color</b>	Defines the cell background color.
<b>Text Color</b>	Defines the cell text color.
<b>Border before cell</b>	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.
<b>Border after cell</b>	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.
<b>Text settings</b>	<p>In this group you find a number of text modifiers applicable to the cell.</p> <p><b>Text Size</b> Use the drop-down to apply a size modification to the general font used in the table.</p> <p><b>Bold</b> Mark this check box for bold text.</p> <p><b>Italic</b> Mark this check box for italic text.</p> <p><b>Underlined</b> Mark this check box for underlined text.</p> <p><b>Drop Shadow</b> Mark this check box for text with drop shadow.</p>
<b>Apply change to</b>	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.

### 36.3 Table Box Properties: General

The **Table Box Properties: General** page is opened by a right-click on a table box and choosing the **Properties** command from the float menu, or by selecting **Properties** in the **Object** menu when a table box is active.

In the **Title** window the table box can be given a name which will be displayed in the window caption. The title can also be defined as a *Calculated Formula* (page 871) for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.



*Table Box Properties, General*

#### **Available Fields**

In this column the names of the data source fields are shown. Initially all fields (excluding System fields) appear in this column. To include the system fields, check **Show System Fields**. Select the items to be used/removed by clicking them. Use the **Add >** or the **< Remove** button to move them to the desired column.

Provided that the script has been executed in QlikView version 5.03 or later, key fields will be indicated with a key symbol.

#### **Fields Displayed in Tablebox**

In this column the names of the fields selected from the **Available Fields** list to be included in the table box are shown. Initially no fields are in this column.

When a field is selected in this list, you may give it a custom **Label** in the edit window.

#### **Show System Fields**

Check this box to make the system fields appear in the **Available Fields** column.

#### **Show Fields from Table**

From here, you control what fields appear in the **Available Fields** list. The drop-down list displays the alternative **All Tables** by default. If you want the list to show fields from a specific table, select the table name in the drop-down list.

The alternative **All Tables (Qualified)** shows the fields qualified by the name of the table(s) they occur in. This means that key (connecting) fields will be listed more than once. (This alternative is only used for viewing purposes and has nothing to do with *Qualify* (page 264) fields in the load script.)

#### **Promote**

Moves a field upwards in the display order.

#### **Demote**

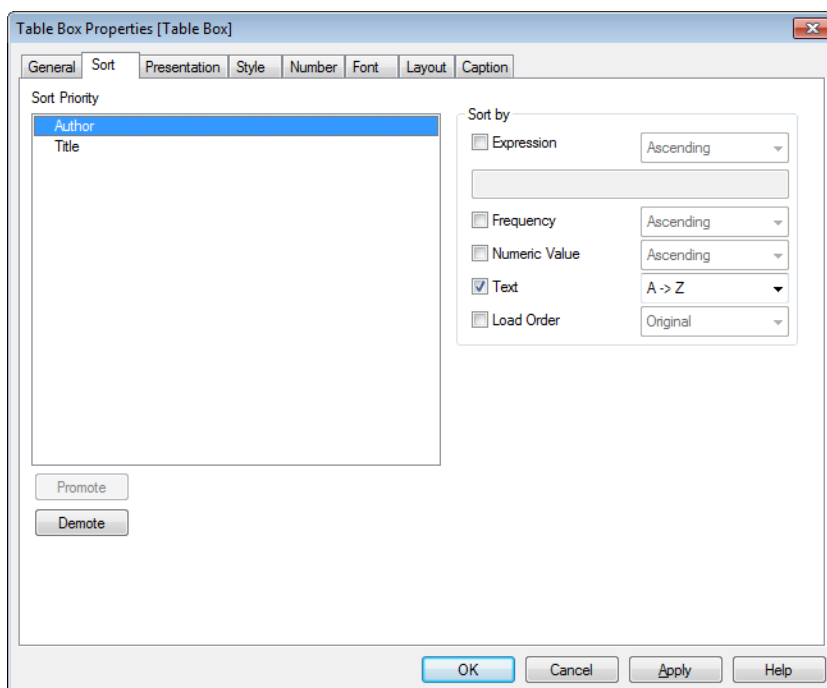
Moves a field downwards in the display order.

#### **Count Order**

Sorts the fields in the **Fields Displayed in Tablebox** column in numeric order.

<b>Load Order</b>	Sorts the fields in the <b>Fields Displayed in Tablebox</b> column in load order, i.e. the order in which they are read from the database.
<b>Text Order</b>	Sorts the fields in the <b>Fields Displayed in Tablebox</b> column in alphabetical order.
<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .
<b>Object ID</b>	This is used for <i>Internal Macro Interpreter</i> (page 893) purposes. Every sheet object is assigned a unique ID, which for table boxes start with TB01. Shared sheet objects share the same object ID. It is possible to edit this ID number later on.
<b>Calculation Condition</b>	Specify a <b>Calculation Condition</b> which must be fulfilled for the table box to be displayed. As long as the condition is not met, the message "Calculation condition unfulfilled" will be displayed instead.
<b>Error Messages</b>	The standard error messages in table boxes (and charts) can be customized in the <i>Custom Error Messages</i> (page 875) dialog which is opened from the <b>Error Messages</b> button.

## 36.4 Table Box Properties: Sort



*Table Box Properties, Sort*

The **Table Box Properties: Sort** tab is opened by a right-click on a table box and choosing the **Properties** command from the float menu.

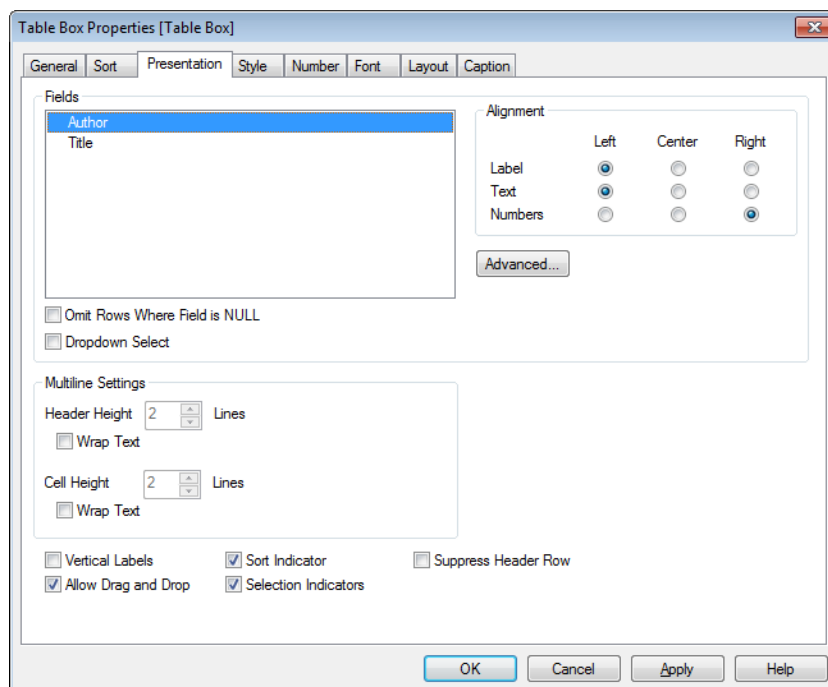
In the **Sort Priority** list the fields of the table box are displayed. The order of the fields determine the sort order applied when the **Sort** command is executed. The order of the fields can be changed by using the **Promote** and **Demote** buttons. For each field in the list the criteria to be used in the sorting procedure can be specified. The sort criteria are described below.

**Sort by:**

<b>Expression</b>	Field values will be sorted according to an arbitrary expression that is entered into the text edit box below this sort option.
<b>Frequency</b>	Toggles the status whether or not the sorting procedure of values should be made by frequency.
<b>Numeric value</b>	Toggles the status whether or not the sorting procedure of values should be made by numeric value.
<b>Text</b>	Toggles the status whether or not the sorting procedure of values should be made in alphabetical order according to the ASCII standard.
<b>Load order</b>	Toggles the status whether or not the sorting procedure of values should be made in load order.

If more than one sort order is specified, the sort order will be expression, frequency, numeric, text, load order.

## 36.5 Table Box Properties: Presentation

*Table Box Properties, Presentation*

The **Table Box Properties: Presentation** tab is opened by a right-click on a table box and choosing the **Properties** command from the float menu.

In this dialog you determine the layout of the table box.



---

<b>Alignment</b>	The default alignment of field values can be set separately for <b>Text</b> and <b>Numbers</b> by selecting the field in the <b>Fields</b> list and the appropriate option in the <b>Alignment</b> group. As default, text values are left aligned, numbers values are right aligned.
<b>Omit Rows Where Field is NULL</b>	If checked, rows containing a field value that is NULL from the selected field are removed. By NULL "no value" is understood.
<b>Dropdown Select</b>	If checked, a dropdown arrow icon is added to the left side of the column header of the selected field. Click the icon to access the field values from a dropdown field list. This works exactly like making selections in a multi box.
<b>Advanced...</b>	This button opens the <i>Advanced Field Settings (page 515)</i> dialog which offers settings for image representation of field values and special text search options.
<b>Multiline Settings</b>	<p>In this group the table header and data cells may be set to display values in multiple rows, which is useful for long text strings.</p> <p><b>Wrap TextHeader Height <i>n</i> Lines</b> With this option selected, the header will display its contents in more than one row. Here the desired limits to the number of header lines is specified.</p> <p><b>Wrap TextCell Height <i>n</i> Lines</b> With this option selected, a cell will display its contents in more than one row. Here the desired limits to the number of cell row lines is specified.</p>
<b>Vertical Labels</b>	If checked, all field labels will be rotated to the vertical position.
<b>Allow Drag and Drop</b>	If checked, it will be possible to sort the table box fields by clicking and dragging the headers.
<b>Sort Indicator</b>	If checked, a sort indicator icon is added to the right side of the column header of the field that the table box is currently sorted by. The icon is flipped to reflect ascending or descending sort order.
<b>Selection Indicators</b>	If checked, a sort indicator icon is added to the right side of the column header of the field that the table box is currently sorted by. The icon is flipped to reflect ascending or descending sort order.
<b>Suppress Header Row</b>	If checked, the table will be displayed without header (label) row.

## 36.6 Chart Properties: Style

Current Style  
Light

Stripes every 0 Rows

Cell Background Color Transparency  
0 % 100 %

Cell Borders Transparency  
0 % 100 %

Border Above Spacing

Background...

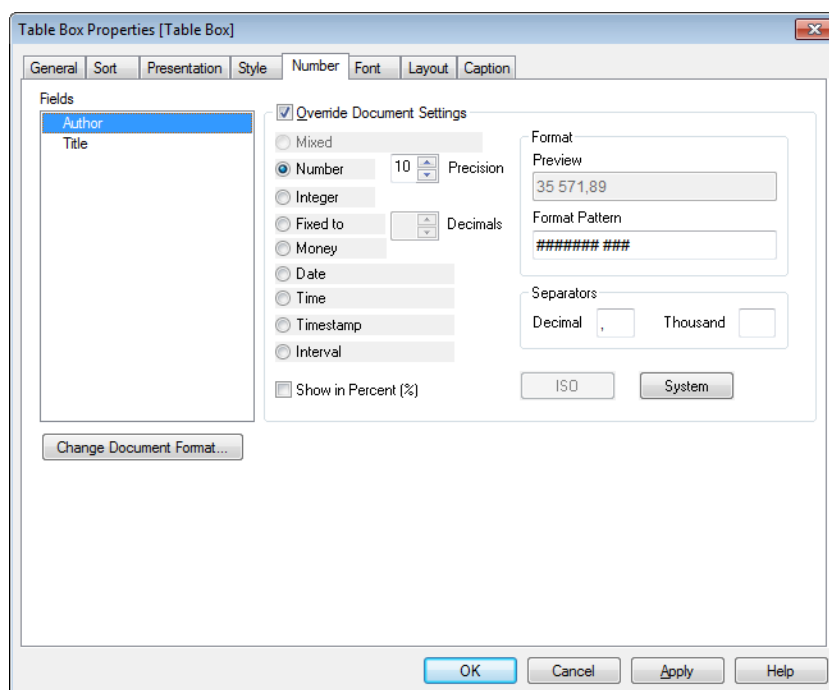
### *Chart Properties, Style*

This style page applies to all QlikView tables; table boxes, pivot tables and straight tables. Here you make settings for the table formatting style.

<b>Current Style</b>	Choose an appropriate table style from the drop-down list. If the value [ <b>Custom</b> ] 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.
<b>Stripes every _ Rows</b>	Here you can specify if and at how long intervals shaded stripes should appear.
<b>Indent Mode</b>	This setting is only valid for pivot tables. With this alternative checked, you can achieve a slightly different table style that is especially useful when you need to accommodate a number of dimension labels within a limited table width. <b>Use Only First Dimension Label</b> This setting is only available for pivot tables already in <b>Indent Mode</b> and modifies the style of the pivot table further.
<b>Vertical Dimension Cell Borders</b>	This setting determines whether vertical cell borders are displayed for dimension columns.
<b>Vertical Expression Cell Borders</b>	As above, but for expression columns.
<b>Border Above Spacing</b>	Provided that a <b>Spacing</b> has been determined in the <i>Advanced Field Settings</i> (page 515) dialog, the table style can be slightly modified by checking this alternative.
<b>Background...</b>	Opens the <i>Background Settings</i> (page 489) dialog.
<b>Cell Background Color Transparency</b>	If a color or an image has been applied in <b>Background Settings</b> , you can adjust the transparency of that color or image in the cell background here.

**Cell Borders Transparency** Sets how pronounced the cell borders should be.

## 36.7 Table Box Properties: Number



*Table Box Properties, Number*

The **Table Box Properties: Number** tab is opened by a right-click on a table box and choosing the **Properties** command from the float menu.

This property page provides formatting settings for all fields of the table box. The number formatting can be set individually for fields by selecting one or several fields (Click, Shift-click or Ctrl-click) in the **Fields** text box.

Each field has a default number format which can be set in the *Document Properties: Number (page 449)* page. It is however possible to use a separate number format for an individual sheet object. To do this, check the alternative **Override Document Settings** and specify a number format in the group control below. This property page applies to the active object and contains the following controls for formatting values:

<b>Mixed</b>	Both numbers and text. Numbers shown in original format.
<b>Number</b>	Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.
<b>Integer</b>	Shows numeric values as integers.
<b>Fixed to _ Decimals</b>	Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.
<b>Money</b>	Shows values in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.
<b>Date</b>	Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.

<b>Time</b>	Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Timestamp</b>	Shows values that can be interpreted as date + time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Interval</b>	Shows time as sequential time increment (e.g. format = <i>mm</i> shows the value as the number of minutes since calendar start ( 1899:12:30:24:00)).

The **Show in Percent (%)** button operates on the following formats: **Number**, **Integer** and **Fixed to**.

**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

The **ISO** button uses the **ISO** standard for the formatting of date, time and timestamp.

The **System** button applies the settings of the system to the formatting.

The **Change Document Format** button opens the *Document Properties: Number (page 449)* page where the default number format of any field can be edited.

## 36.8 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 36.9 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 36.10 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 37 Current Selections Box

Current Selections	
Fields	Values
Country	<input checked="" type="radio"/> Germany
Customer	<input checked="" type="radio"/> Atlantic Marketing


In the current selections box, selections are listed by field name and field value. This tool displays the same information as the free-floating *Current Selections* (page 103) window but it is positioned directly on the sheet like any other sheet object. The *Indicators* (page 101) is used for distinguishing between selected and locked values.



By a right-click on a current selections box the *Current Selections Box: Object menu* (page 533) will be displayed. It can also be accessed from the **Object** menu, when the current selections box is the active object.

### 37.1 Current Selections Box: Object menu

The current selections box **Object** menu has two different versions. If the caption area is clicked the float menu will contain commands that are applicable to the whole box, if a specific entry is under the pointer the menu will be extended to include commands that apply to that entry. The commands are:

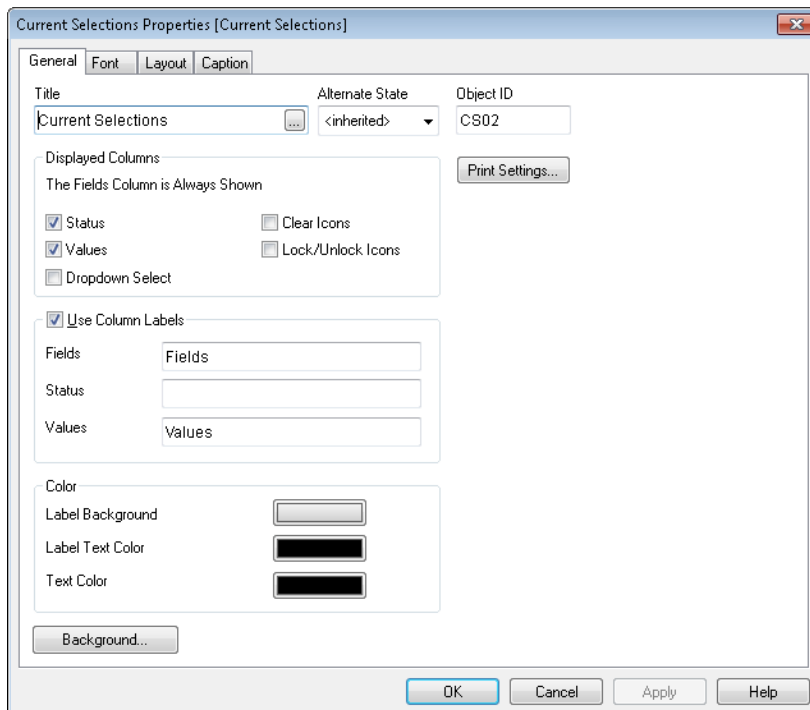
<b>Properties...</b>	Opens the <b>Current Selections Box Properties</b> dialog where <i>Current Selections Box Properties: General</i> (page 535), <i>Font</i> (page 491) and <i>Layout</i> (page 492) parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Order</b>	This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127. <ul style="list-style-type: none"> <li><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</li> <li><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</li> <li><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</li> <li><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</li> </ul>
<b>Select Possible</b>	All non-excluded values of the field are selected.
<b>Select Excluded</b>	All excluded values of the field are selected.
<b>Select All</b>	All values of the field are selected.
<b>Clear</b>	Clears all the current selections of the field.

<b>Clear other fields</b>	Clears the selections in all the other sheet objects, including selections in other fields of the current selections box, while maintaining selections in this specific field of the current selections box.
<b>Lock</b>	Locks the selected value(s) of the field.
<b>Unlock</b>	Unlocks the locked value(s) of the field.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog where print settings can be specified. All values of the current selections box are sent to the printer.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send to Excel</b>	Exports the text to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. 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.
<b>Export...</b>	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).
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Data</b> Copies the data (selections) in the selected current selections box to the clipboard. <b>Cell Value</b> Copies the text value of the current selections box cell right-clicked upon (when invoking the Object menu) to the clipboard. <b>Image</b> Copies an image of the current selections box to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences</b> dialog, <b>Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.

- Maximize** Enlarges the object 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 object's **Properties** dialog on the *Caption (page 496)* page.
- 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 QlikView help.
- Remove** Removes the sheet object.

## 37.2 Current Selections Box Properties: General

The **Current Selections Box Properties: General** page is opened by a right-click on a current selections box and choosing the **Properties** command in the float menu. Here it is possible to set general parameters of the current selections box.



*Current Selections Box Properties, General*

- Title** The text to appear in the current selections box caption area. The title can also be defined as a *Calculated Formula (page 871)* for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .
<b>Object ID</b>	This is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for current selections boxes start with CS01. Shared objects share the same object ID. It is possible to edit this ID number later on.

In the **Displayed Columns** group, you determine if the status and/or the values columns will appear in the current selections box.

<b>Status</b>	Checking this box will add a <b>Status</b> column with indicator to the current selections box.
<b>Values</b>	Checking this box will add a <b>Values</b> column to the current selections box, listing selected field value.
<b>Drop-down Select</b>	Enable this setting to display a drop-down icon for each field in the current selections box, thus making it possible to modify selections within the object.
<b>Clear Icons</b>	If this check box is marked each field row in the current selections box will display a small clear icon. Clicking on the clear icon will clear selections in the field. No clear icon will be shown for locked fields.
<b>Lock/Unlock Icons</b>	If this check box is marked each field row in the current selections box will display a small lock or unlock icon. Clicking on the icon will lock or unlock selections in the field.

Checking **Use Column Labels** enables the following settings:

<b>Fields</b>	The label to be shown above the <b>Fields</b> column can be edited in the text box.
<b>Status</b>	The label to be shown above the <b>Status</b> column can be edited in the text box.
<b>Values</b>	The label to be shown above the <b>Values</b> column can be edited in the text box.

In the **Color** group you may edit the colors of the different components of the current selections box.

<b>Label Background</b>	Defines the background color of the label row.
<b>Label Text Color</b>	Defines the text color of the label row.
<b>Text Color</b>	Defines the text color of the display area.
<b>Background...</b>	Opens the <i>Background Settings (page 489)</i> dialog.



## 37.3 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 37.4 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 37.5 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



## 38 Input Box

Forecasted increase		
Increase%	=	10

The input box is a sheet object that is used for entering data into QlikView variables and displaying their values.

Right-click on the input box to display the *Input Box: Object Menu* (page 539). It can also be accessed from the **Object** menu, when the input box is the active object.

### 38.1 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 *Fields* (page 171).

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.

### 38.2 Input Box: Object Menu

By clicking once with the right mouse button while pointing on an input box, a float menu appears. This menu can also be found under **Object** in the Main menu when an input box is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Input Box Properties</b> page where its parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.

- Order** This cascade menu is only available when the **Design Grid** command of the *View Menu* (page 52) is activated or when the *Always Show Design Menu Items* (page 78) checkbox is marked. 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 *Print: General* (page 91) dialog where print settings can be specified. All values of the current selections box are sent to the printer.
- Print as PDF...** Opens the **Print** dialog with the QlikViewPDF printer preselected. After pressing the **Print** button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
- Send to Excel** Exports the input box values to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. The exported values 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).
- Copy to Clipboard** This cascade 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: 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.

**Linked Objects**

Opens a menu with the following commands for linked objects.


**Adjust Position of Linked Objects**

All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.


**Unlink This Object/Unlink Objects**

This destroys the link between the objects, making them different objects with different object IDs.


**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 object's **Properties** dialog on the *Caption (page 496)* page.

**Maximize**

Enlarges the object 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 object's **Properties** dialog on the *Caption (page 496)* page.

**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**

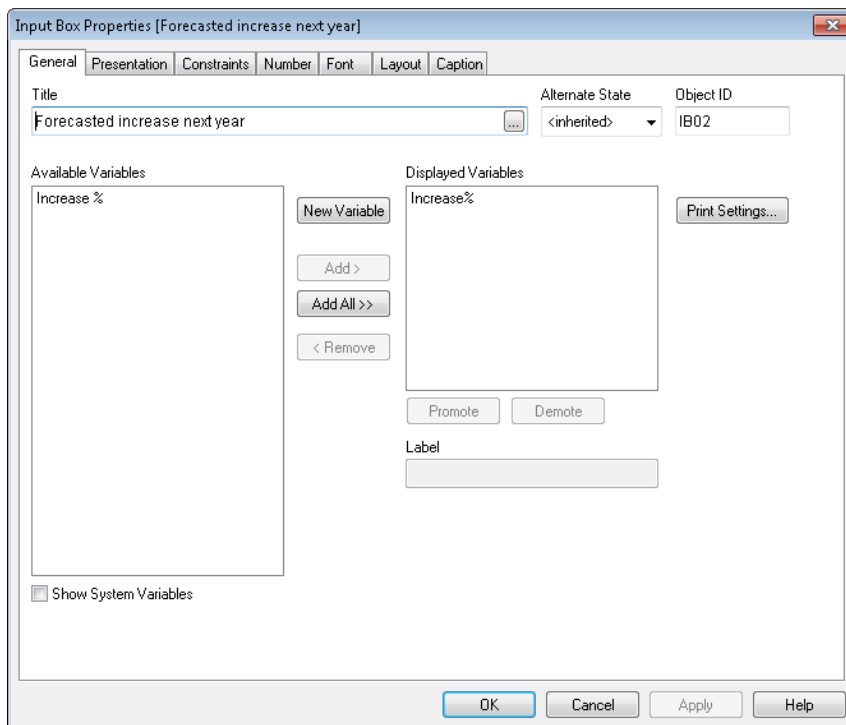
Opens QlikView help.

**Remove**

Removes the sheet object.

## 38.3 Input Box Properties: General

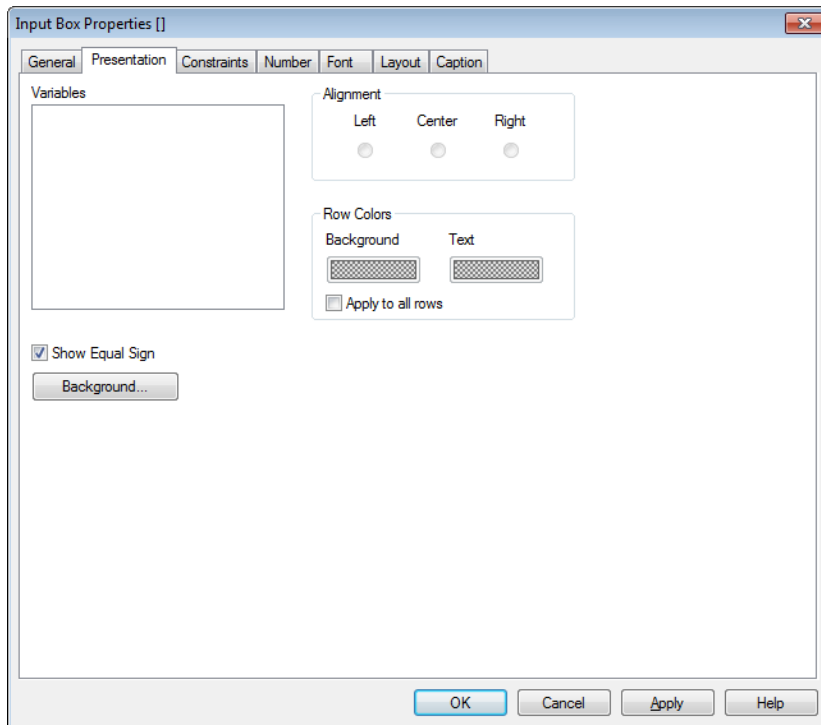
The **Input Box Properties: General** page is opened by a right-click on an input box and choosing the **Properties** command in the float menu. Here it is possible to set general parameters of the input box.



*Input Box Properties, General*

<b>Title</b>	The text to appear in the input box caption area. The title can be defined as a <i>Calculated Formula (page 871)</i> for dynamic update of the label text. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas.
<b>Available Variables</b>	All available variables are listed in this column. To include the system variables, mark the <b>Show System Variables</b> check box. Select the items to be used/removed by clicking them. Use the <b>Add &gt;</b> or the <b>&lt; Remove</b> button to move them to the desired column.
<b>Show System Variables</b>	Makes the system variables show in the list of <b>Available Variables</b> .
<b>New Variable</b>	Opens the <b>New Variable</b> dialog where a custom variable may be defined.
<b>Displayed Variables</b>	Variables to be displayed in the input box are listed in this column. Initially the column is empty.
<b>Promote</b>	Moves a variable upwards in the display order.
<b>Demote</b>	Moves a variable downwards in the display order.
<b>Label</b>	An alternative name to be used as variable title in the input box, can be entered here. The label may be defined as a <i>Calculated Formula (page 871)</i> for dynamic update. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas.
<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .
<b>Object ID</b>	This is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for input boxes start with IB01. Linked sheet objects share the same object ID. You may edit this ID number later on.

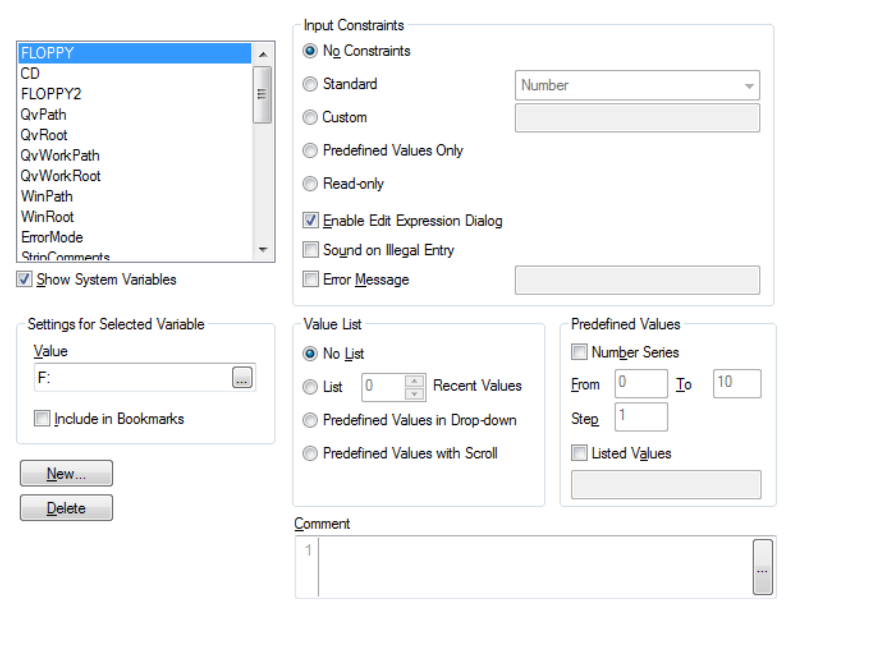
## 38.4 Input Box Properties: Presentation



*Input Box Properties, Presentation*

- 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.
- Background...** Opens the *Background Settings (page 489)* dialog.
- Alignment** The alignment of the variables can be set. The variables can be individually set to left, center or right aligned.
- Row Colors** In this group individual color settings can be made for the row selected in the **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.

## 38.5 Input Box Properties: Constraints and Document Properties: Variables



*Input Box Properties, Constraints and Document Properties, Variables*

The **Input Box Properties: Constraints** page is opened by right-clicking on an input box and choosing the **Properties** command in the float menu. The *Variable* (page 173) that have been added to **Displayed Variables** in the *Input Box Properties: General* (page 541) page are listed in the **Variables** list on the **Constraints** tab where it can be modified.

The **Document Properties: Variables** page is accessed from **Settings - Document Properties: Variables**. The list of variables may be modified by the **Show System Variables** option. The **New...** button adds a new variable to the document. **Delete** button removes the selected variable.

In the **Settings for Selected Variable** group the edit box displays the current **Value** of the selected variable. The value may be entered as a calculated formula. Click on the **...** button to open the *Edit Expression Dialog* (page 757).

Variables are normally not included in bookmarks, but here it is possible to check the option **Include in Bookmarks**.

In the **Input Constraints** group 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 may be shown. The following alternatives are available:

- No Constraints**      Input values are not checked against any specified constraints.
- Standard**            Input values will be checked against one of a number of common standard constraints selected from the drop-down box. As default no constraint at all is selected, i.e. any value can be entered into the variable. Only one of the alternatives: **Standard**, **Custom**, **Predefined Values Only** and **Read-only** can be checked.



<b>Custom</b>	<p>Input values will be checked against a user specified constraint. The constraint is entered in the edit box and must 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.</p> <p>Example:</p> <p><math>\\$ &gt; 0</math> will cause the input box to accept only positive numbers in the selected variable.</p> <p>The previous value of the variable can be referenced by the variable name.</p> <p>Example:</p> <p><math>\\$ \geq abc + 1</math> 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.</p>
<b>Predefined Values Only</b>	Input values will be checked against a list of values defined in the <b>Predefined Values</b> group. Only input values found in the list will be accepted.
<b>Read-only</b>	Marks the variable as read-only. No values can be entered.
<b>Enable Edit Expression Dialog</b>	Check this alternative to enable editing of the variable value in the <b>Edit Expression</b> dialog, which is opened from the ... button that appears when clicking on the value itself.
<b>Sound on Illegal Entry</b>	Check this alternative to have QlikView issue a sound warning when the user attempts to enter a value which falls outside of the constraints.
<b>Error Message</b>	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 checking this alternative it is possible to specify a <i>Custom Error Messages</i> (page 875) 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 <i>Calculated Formula</i> (page 871) for dynamic update. Click the ... button to open the <b>Edit Expression</b> dialog for easier editing of long formulas.

In the **Value List** group it is specified if and how to present a list of values for an input box.

<b>No List</b>	No list of values previously used in the variable is maintained.
<b>List _ Recent Values</b>	A drop-down (MRU) list with the values most recently used, 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 <b>entries</b> box.
<b>Predefined Values in Drop-down</b>	A drop-down list with <b>Predefined Values</b> will be available to the user for the selected variable in the input box.
<b>Predefined Values with Scroll</b>	A scroll control will be available to the user for the selected variable in the input box. The scrolling will be made between the <b>Predefined Values</b> .

In the **Predefined Values** group a list of predefined values is specified, which can be presented to the user in a drop down and/or used for defining acceptable variable values.

<b>Number Series</b>	Check this alternative to generate a list of predefined numeric values based on a lower limit ( <b>From</b> ), an upper limit ( <b>To</b> ) and a <b>Step</b> value. This option can be used alone or in combination with <b>Listed Values</b> .
----------------------	--

<b>Listed Values</b>	Check this alternative to specify a list of arbitrary predefined values. The values can be either numeric or alphanumeric. Alphanumeric values must be enclosed in quotes (e.g. 'abc'). Each value is separated by a semicolon ( ; ) (e.g. 'abc';45;14.3;'xyz' ). This option can be used alone or in combination with <b>Number Series</b> .
<b>Comment</b>	This is a commentary field where the creator of a variable can describe its purpose and function.

## 38.6 Input Box Properties: Number

The **Input Box Properties: Number** page is opened by a right-click on an input box and choosing the **Properties** command from the float menu.

This property page provides formatting settings for all variables of the input box. The number formatting can be set individually for variables by selecting one or several variables (Click, Shift-click or Ctrl-click) in the **Variables** text box.

Each variable has a default number format which can be set in the *Document Properties: Number* (page 449) page. It is however possible to use a separate number format for an individual sheet object. To do this, check the alternative **Override Document Settings** and specify a number format in the group control below. This property page applies to the active object and contains the following controls for formatting variable values:

<b>Mixed</b>	Both numbers and text. Numbers shown in original format.
<b>Number</b>	Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.
<b>Integer</b>	Shows numeric values as integers.
<b>Fixed to _ Decimals</b>	Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.
<b>Money</b>	Shows values in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.
<b>Date</b>	Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Time</b>	Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Timestamp</b>	Shows values that can be interpreted as date + time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.
<b>Interval</b>	Shows time as sequential time increment (e.g. format = <i>mm</i> shows the value as the number of minutes since calendar start ( 1899:12:30:24:00).

The **Show in Percent (%)** button operates on the following formats: **Number**, **Integer** and **Fixed to**.

**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

The **ISO** button uses the **ISO** standard for the formatting of date, time and timestamp.

The **System** button applies the settings of the system to the formatting.

## 38.7 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 38.8 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 38.9 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



## 39 Button



Buttons can be created and defined in QlikView to perform commands or actions. There are basically three types of buttons; the **Launch/Export Button**, the **Shortcut Button** and the **Macro** button.

By right-clicking the object, the *Button: Object Menu* (page 549) will be displayed. It can also be accessed from the **Object** menu, when the button is the active object.

### 39.1 Button: Object Menu

The button **Object** menu is opened by right-clicking the object. The menu commands are:

<b>Properties...</b>	Opens the <i>Button Properties: General</i> (page 550) page where you can set the parameters defining the button.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Send to Excel</b>	Exports the text to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. 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.
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Text</b> Copies the text displayed on the button to the clipboard.</p> <p><b>Image</b> 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 <b>User Preferences</b> dialog, <b>Export</b> page.</p> <p><b>Object</b> 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.</p>

**Linked Objects**

Opens a menu with the following commands for linked objects.


**Adjust Position of Linked Objects**

All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.


**Unlink This Object/Unlink Objects**

This destroys the link between the objects, making them different objects with different object IDs.


**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 object's **Properties** dialog on the *Caption* (page 496) page.

**Maximize**

Enlarges the object 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 object's **Properties** dialog on the *Caption* (page 496) page.

**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.

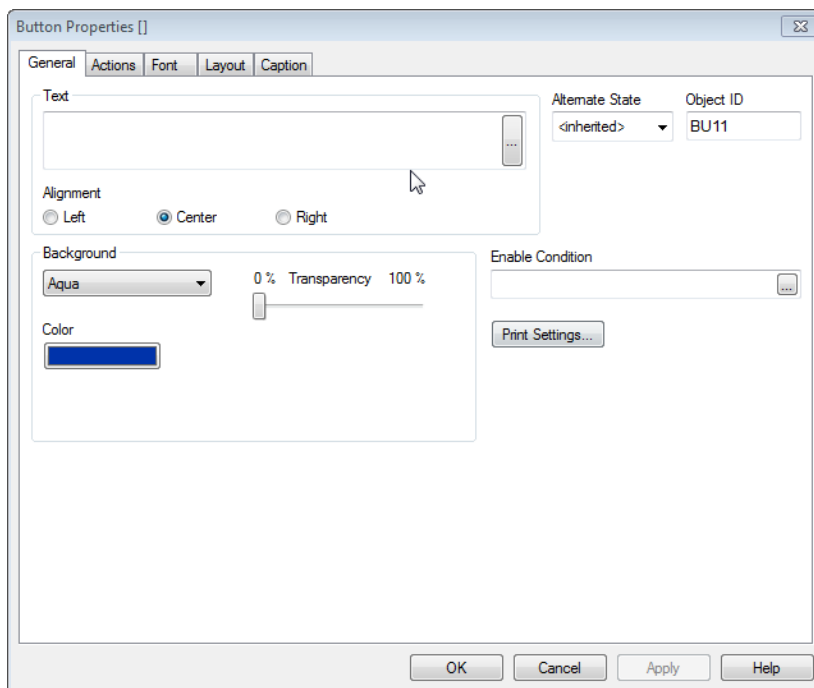
**Help**

Opens QlikView help.

**Remove**

Removes the sheet object.

## 39.2 Button Properties: General



*Button Properties, General*

In the **Text** group you can set the properties of the text to be displayed on the button.

**Text** Entering a text in the text field labels the button. The text may be defined as a *Calculated Formula* (page 871). The text may consist of several lines, divided when you press Enter.

**Alignment** The button text can be left-aligned, centered or right-aligned.

In the **Background** group you can specify the background of the button. The drop-down gives you three different basic button styles to choose from:

**Aqua** Default for new buttons. Gives a rounded button with three-dimensional glassy look.

**Plain** Produces a traditional plain QlikView button.

**System Default**

Provides a solid background in the color defined for buttons in the operating system.

**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.

**Color** Select this radio button if you want the button to be displayed with a colored background (not available with the Image setting). 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.

**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.

**Alternate State** Choose one of the available states in the list. The following Alternate States are always available.

**Inherited**

The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.

**Default state**

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

<b>Object ID</b>	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. Shared objects share the same object ID. 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.
<b>Enable Condition</b>	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.
<b>Print Settings...</b>	By pressing this button you will gain access to the Print Settings dialog where margins and header/footer format can be defined. The <b>Print Settings</b> dialog holds two pages, <i>Print: Layout (page 92)</i> and <i>Print: Header/Footer (page 93)</i> , which are identical to the last two pages of the <b>Print</b> dialog.

## 39.3 Actions

On this page actions can be set for certain objects:

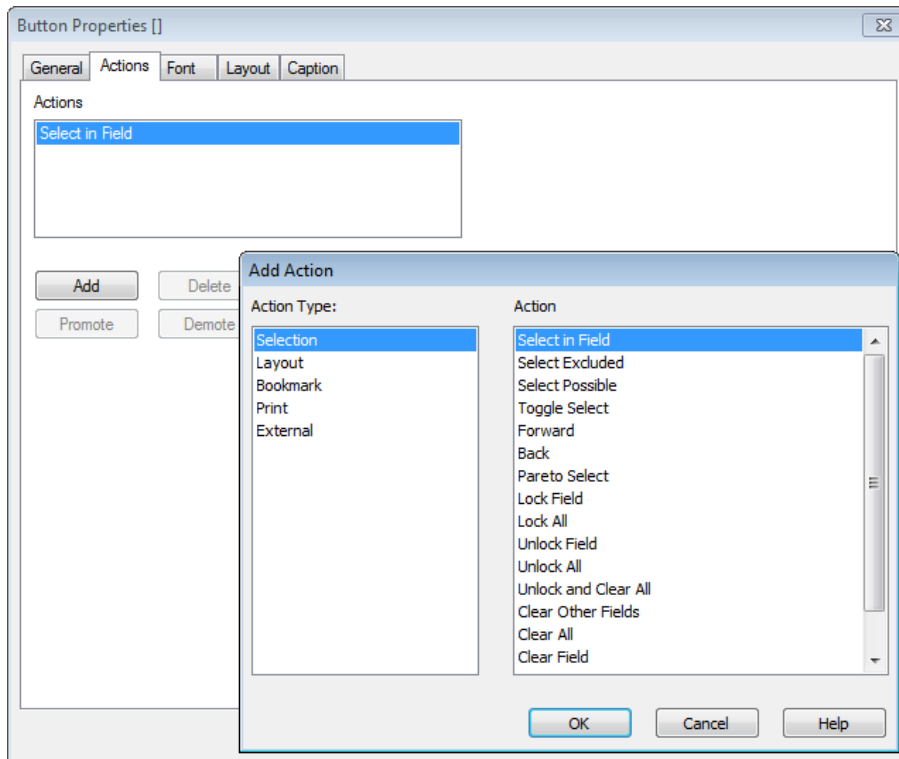
- buttons
- text objects
- gauges
- line/arrow objects

Actions include launching an external application or exporting data from QlikView.

<b>Add</b>	Opens the <i>Add Action Dialog (page 553)</i> where the actions for the object are chosen. Choose the type of action in the lists. Depending on the action, different parameters for the action will appear on the <b>Actions</b> page. The actions will be executed in the order in which they appear in the list.
<b>Delete</b>	Removes the action from the object.
<b>Promote</b>	Moves the selected action up in the list of actions.
<b>Alternate State</b>	Sets the state that the action refers to. This setting is only relevant for actions related to selections or bookmarks.



## Add Action Dialog



*The Actions page*

The following actions can be added from the **Add Action** dialog. In all fields with the ... button it is possible to enter a calculated formula, see *Calculated Formula (page 871)*.

## Selection

<b>Select in Field</b>	Selects the values and fields that are specified. In the <b>Search String</b> you can specify a search mask, e.g. (A B) will select both A and B.
<b>Select Excluded</b>	Selects the excluded values in the specified field.
<b>Select Possible</b>	Selects the possible values in the specified field.
<b>Toggle Select</b>	Toggles between the current selection and the specified <b>Field</b> and <b>Search String</b> . In the <b>Search String</b> you can specify a search mask, e.g. (A B) will select both A and B.
<b>Forward</b>	Goes one step forward in the list of selections.
<b>Back</b>	Goes back one step in the list of selections.
<b>Pareto Select</b>	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, <i>Customer</i> should be used as field, <i>sum(Turnover)</i> should be used as expression and 80 should be used as percentage.
<b>Lock Field</b>	Locks the selections in the specified field.
<b>Lock All</b>	Locks all values in all fields.
<b>Unlock Field</b>	Unlocks the selections in the specified field.
<b>Unlock All</b>	Unlocks all values in all fields.
<b>Unlock and Clear All</b>	Unlocks all values and clears all selections in all fields.
<b>Clear Other Fields</b>	Clears all related fields except the one specified.
<b>Clear All</b>	Clears all selections except locked ones.
<b>Clear Field</b>	Clears a specific field.
<b>Copy State Contents</b>	Copies the selection from the <b>Source State</b> to the <b>Target State</b> . Only available when alternate states have been defined.
<b>Swap State Contents</b>	Swaps the selections between <b>State 1</b> and <b>State 2</b> . Only available when alternate states have been defined.

---

## Layout

<b>Activate Object</b>	Activates the object specified by <b>Object ID</b> . This function does not work in Ajax client.
<b>Activate Sheet</b>	Activates the sheet specified by <b>Sheet ID</b> .
<b>Activate Next Sheet</b>	Opens the next sheet in the document.
<b>Activate Previous Sheet</b>	Opens the previous sheet in the document.
<b>Minimize Object</b>	Minimizes the object specified by <b>Object ID</b> .
<b>Maximize Object</b>	Maximizes the object specified by <b>Object ID</b> .
<b>Restore Object</b>	Restores the object specified by <b>Object ID</b> .
<b>Set State Name</b>	Applies the specified <b>State</b> to the object specified by <b>Object ID</b> . Only available when having defined alternate states.

## Bookmark

<b>Apply Bookmark</b>	Applies a bookmark specified by <b>Bookmark ID</b> . If two bookmarks have the same ID, the document bookmark is applied. To apply the server bookmark specify Server\bookmarkID.
<b>Create Bookmark</b>	Creates a bookmark from the current selection. Specify <b>Bookmark ID</b> and <b>Bookmark Name</b> . Select <b>Hidden</b> to create a hidden bookmark.
<b>Replace Bookmark</b>	Replaces the bookmark specified by <b>Bookmark ID</b> with the current selection.

## Print

<b>Print Object</b>	Prints the object specified by <b>Object ID</b> . Specify the <b>Printer name</b> if the object should be sent to another printer than the default printer. (Not available as document and sheet trigger.)
<b>Print Sheet</b>	Prints the sheet specified by <b>Sheet ID</b> . (Not available as document and sheet trigger.) This function does not work in AJAX client.
<b>Print Report</b>	Prints the report specified by <b>Report ID</b> . Specify the <b>Printer name</b> if the report should be sent to another printer than the default printer. Mark the check box <b>Show Print Dialog</b> if you want the Windows print dialog to be shown. (Not available as document and sheet trigger.)

## External

<b>Export</b>	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 <b>Setup</b> button on the <b>Actions</b> page to open the <b>Export Action Settings</b> dialog. (not available as document and sheet trigger)
<b>Launch</b>	<p>Launches an external program. The following settings can be configured in the <b>Actions</b> dialog:</p> <p><b>Application</b> Click on <b>Browse...</b> to find the application that should be launched. (Not available as document and sheet trigger.)</p> <p><b>Filename</b> Enter the path to the file that should be opened with the application specified above. (Not available as document and sheet trigger.)</p> <p><b>Parameters</b> Specify parameters for the command line from which the application is started. (Not available as document and sheet trigger.)</p> <p><b>Working Directory</b> Sets the working directory for the application to be launched. (Not available as document and sheet trigger.)</p> <p><b>Exit application when QlikView is closed</b> Forces the application to be closed when QlikView is exited. (Not available as document and sheet trigger.) This function does not work in AJAX client.</p>
<b>Open URL</b>	Opens the URL in the default web browser. (Not available as document and sheet trigger.)
<b>Open QlikView Document</b>	<p>Opens the specified document. The file extension must be included. (Not available as document and sheet trigger.)</p> <p>Mark the check box <b>Transfer State</b> to transfer the selections from the original document to the one you wish to open. The opened document will first be cleared of selections.</p> <p>Mark <b>Apply state on top of current</b> to retain the second document's selections and apply the original document's selections on top of them.</p> <hr/> <p><b>Note!</b> Use <b>Apply state on top of current</b> with caution, as it can cause unpredictable results if the two documents selections are conflicting. In most cases it should be sufficient with using the <b>Transfer State</b> option.</p> <hr/> <p><b>Open in same Window</b> opens the new document in the same browser tab when using the AJAX ZFC client.</p> <hr/> <p><b>Note!</b> Open QlikView Document action is not supported when using the Internet Explorer plugin</p> <hr/>
<b>Run Macro</b>	Enter the path and name of the macro to be run. Type any name for which it is possible to later create a macro in the <b>Edit Module</b> dialog, or a <b>calculated expression</b> for dynamic update.
<b>Set Variable</b>	Assigns a value to the specified variable.
<b>Show Information</b>	Shows the associated information, such as a text file or an image for the field specified by <b>Field</b> . This function does not work in AJAX client.
<b>Close This Document</b>	Closes the active QlikView document.

**Reload**

Performs a reload on the current document. This function does not work in AJAX client.

**Dynamic Update**

Performs a dynamic update of the data in the currently loaded document. The statement for the dynamic update is to be entered in the **Statement** field.

The intended usage of Dynamic Update allows a QlikView Administrator to feed limited amounts of data in to a QlikView document from a single source without running a reload of the document. Analysis can then be performed by multiple clients connecting to the QlikView Server.

**Note!** The uploaded information is stored in RAM only so any data added or updated using Dynamic Update will be lost if a reload of the document is performed.

The following grammar describes the possible statements and their components that can be used with the Dynamic Update feature:

- statements ::= statement { “,” statement }
- statement ::= insert\_statement | update\_statement | delete\_statement | begin\_transaction\_statement | commit\_transaction\_statement
- insert\_statement ::= "INSERT" "INTO" ("\*" | table\_name) field\_list "VALUES" value\_list {" ," value\_list} [{"KEY" ["AUTO" | (" (" field\_list ")")]} [{"REPLACE" (["WITH" "ONE"] | "EACH") }]
- update\_statement ::= "UPDATE" ("\*" | table\_name) set\_clause {" ," | set\_clause} "WHERE" condition [{"AUTO" "INSERT"}]
- delete\_statement ::= "DELETE" "FROM" ("\*" | table\_name) "WHERE" condition
- begin\_transaction\_statement ::= "BEGIN" ("TRANSACTION" | "TRAN") [trans\_name]
- commit\_transaction\_statement ::= "COMMIT" ["TRANSACTION" | "TRAN"] [trans\_name]
- table\_name ::= identifier | quoted\_name
- field\_list ::= "(" field\_name {" ," field\_name} ")"
- value\_list ::= "("value {" ," value} ")"
- set\_clause ::= "SET" field\_name "=" any\_valid\_non\_aggregated\_qlikview\_expression
- field\_name ::= identifier | quoted string
- value ::= identifier | any\_qlikview\_number | quoted string
- condition ::= any\_valid\_non\_aggregated\_qlikview\_expression
- identifier ::= any\_qlikview\_identifier
- quoted\_string ::= "[" [^]+ "]"

**Example:**

```
UPDATE AbcTable SET Discount = 123 WHERE AbcField=1
```

**Note!** To use this feature, Dynamic Update must be allowed both on the Document and on the Server.

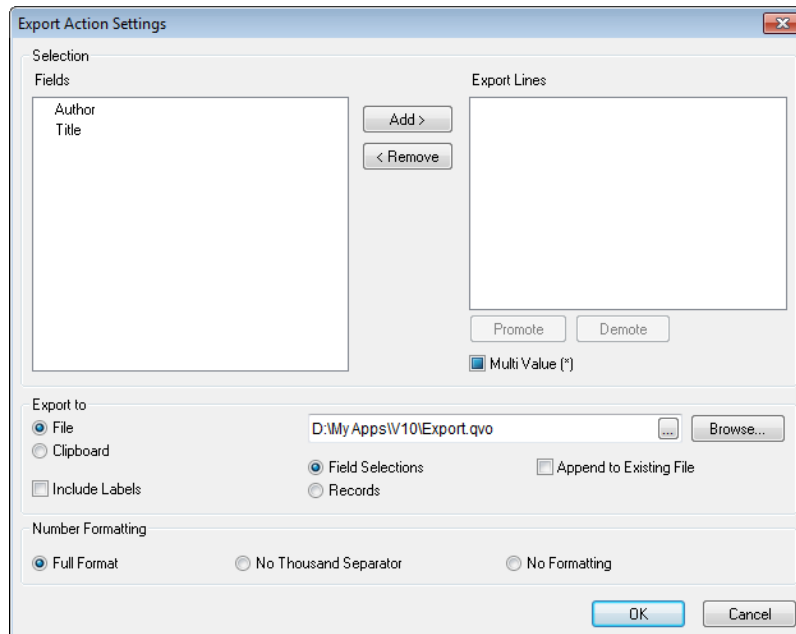
**Note!**

Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

**Note!**

There are limitations as to the behavior of certain macro triggers when working with documents on QlikView Server. Please study the section *Using Macros in QV Documents on the QV-Server (page 903)* before designing server documents with macro triggers.

## Export Action Settings Dialog



*The Export Action Settings page*

### Selection

In this group you set what fields should be selected for export.

<b>Fields</b>	In the <i>Fields (page 171)</i> box you find a list of the available fields.
<b>Export Lines</b>	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.
<b>Add &gt;</b>	Adds fields to the <b>Export Lines</b> box.
<b>&lt; Remove</b>	Removes fields from the <b>Export Lines</b> box.
<b>Promote</b>	Moves the selected field one step up, i.e. one step to the left in the export table.
<b>Demote</b>	Moves the selected field one step down, i.e. one step to the right in the export table.
<b>Multi Value (*)</b>	By marking a field in the <b>Export Lines</b> 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.

<b>File</b>	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 <i>Calculated Formula (page 871)</i> .
<b>Browse</b>	Opens the <b>Export File</b> dialog, allowing you to browse for a file to which the values should be exported.

<b>Clipboard</b>	Default setting. If no file is specified above, the values will be exported to the clipboard.
<b>Field Selections</b>	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.
<b>Include Labels</b>	If this control is checked, the first position on the line (when <b>Field Selections</b> is set) or the first record (when <b>Records</b> is set) will contain the field names.
<b>Records</b>	If this control is checked, the export file will have one column for each selected field, separated by tabs.
<b>Append to Existing File</b>	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 **User Preferences: Export** page. The setting can however be individually set for each export button.

<b>Full Format</b>	Instructs QlikView to export numeric data with its full number format, just as it is shown in the document's sheet objects.
<b>No Thousand Separator</b>	Removes any thousand separator from numeric data. This option is recommended if data are to be imported into MS Excel.
<b>No Formatting</b>	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 section *Using Macros in QV Documents on the QV-Server (page 903)* before designing server documents with macro triggers.

---

## 39.4 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 39.5 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 39.6 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at

object, document and sheet level.  
See *Caption (page 496)*.



## 40 Text Object




Text objects are used for adding information to the document, such as labels etc. They can be moved around and positioned anywhere in the sheet area, even to areas covered by other sheet objects.

By a right-click on a text object the *Text Object: Object Menu (page 561)* will be displayed. It can also be accessed from the **Object** menu, when the text object is the active object.

### 40.1 Text Object: Object Menu

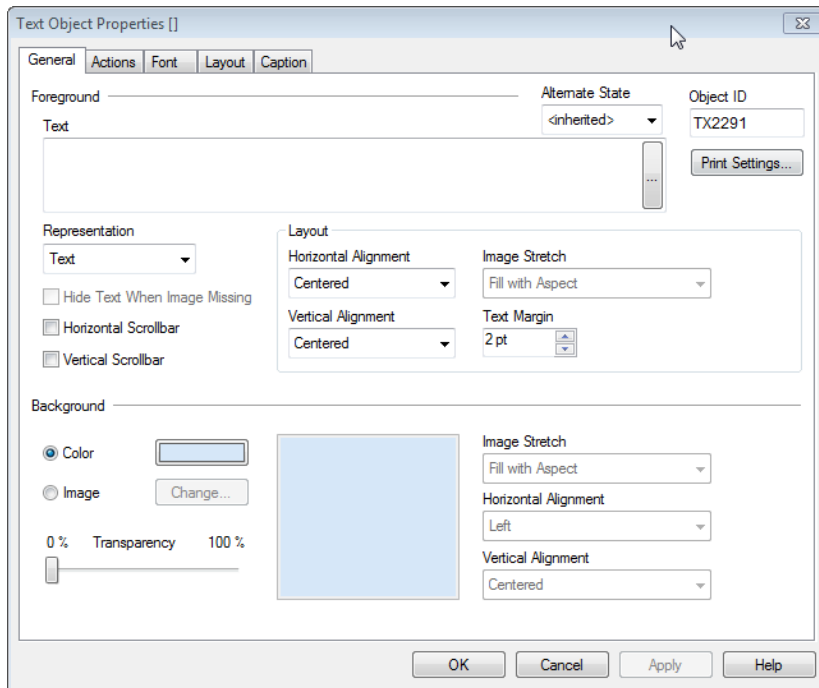
By clicking once with the right mouse button while pointing on a text object, a float menu appears. This menu can also be found under **Object** in the Main menu when a text object is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <i>Text Object Properties: General (page 562)</i> dialog where you can set the parameters defining the text object.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments (page 472)</i> for further information.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog letting you print an image of the text object.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After hitting <b>Print</b> 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.
<b>Send to Excel...</b>	Exports the text to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. 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.

<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Text</b> Copies the text displayed in the text object to the clipboard.</p> <p><b>Image</b> 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 <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	<p>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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Maximize</b>	<p>Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Restore</b>	<p>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.</p>
<b>Help</b>	<p>Opens QlikView help.</p>
<b>Remove</b>	<p>Removes the sheet object.</p>

## 40.2 Text Object Properties: General

The **Text Object Properties: General** page is opened by a right-click on a text object and choosing the **Properties** command from the float menu, or by selecting **Properties** in the **Object** menu when a text object is active. Here settings for the text, its background, alignment etcetera can be made.



*Text Object Properties, General*

### **Foreground**

Here you can specify the foreground of the text object. The foreground usually consists of the text itself, but it can also be an image.

### **Text**

This is where you enter the text to be displayed. The text may be defined as a *Calculated Formula* (page 871) 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 Scrollbar**

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 Scrollbar**

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 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.

**Image Stretch**

This setting describes how QlikView formats a foreground image to fit in the text object area. 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 text object to be filled.

**Fill**

If this option is selected, the 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 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 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* (page 428) 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 transparent. 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. For details on the different options, see the section on **Image Stretch** above.

**Alignment**

The background can be horizontally or vertically aligned.

**Alternate State**

Choose one of the available states in the list. The following Alternate States are always available.

**Inherited**

The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.

**Default state**

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

**Object ID**

The unique ID of the current text object. Upon creation, every sheet object is assigned a unique ID for control via *Internal Macro Interpreter* (page 893). 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. Linked objects share the same object ID. 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: General* (page 91) dialog.

## 40.3 Actions

In the Actions tab 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 *Actions* (page 552).

## 40.4 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 40.5 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 40.6 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

# 41 Line/Arrow Object




Line/arrow objects are used for adding lines or arrows to the layout. They can be moved around and positioned anywhere in the sheet area, even to areas covered by other sheet objects.

By a right-click on a line/arrow object the *Line/Arrow Object: Object Menu (page 567)* will be displayed. It can also be accessed from the **Object** menu, when the line/arrow object is the active object.

## 41.1 Line/Arrow Object: Object Menu

By clicking once with the right mouse button while pointing on a line/arrow object, a float menu appears. This menu can also be found under **Object** in the Main menu when a line/arrow object is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <i>Line/Arrow Object Properties: General (page 568)</i> dialog where you can set the parameters defining the line/arrow object.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments (page 472)</i> for further information.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Image</b> 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 <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>

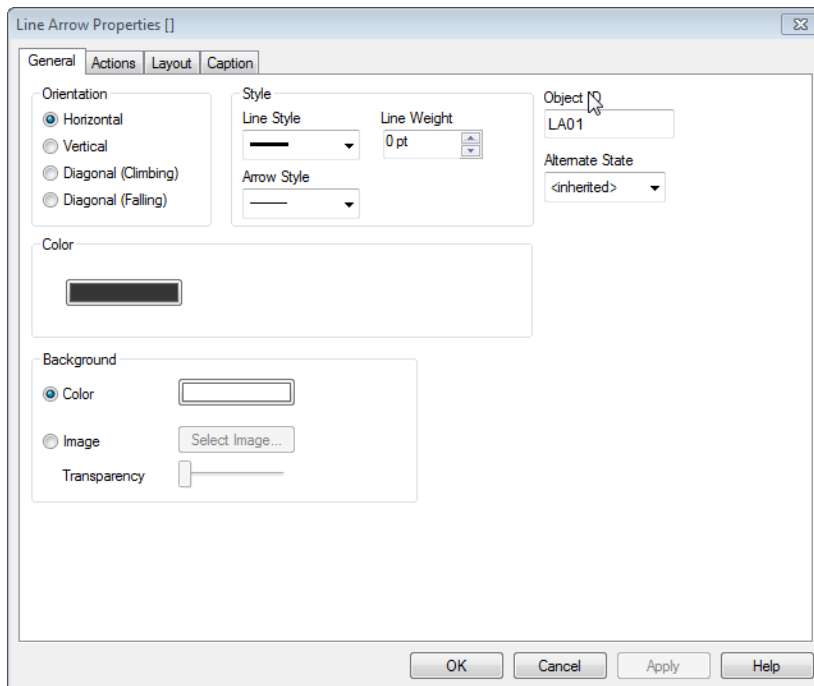
- Minimize** Iconizes the object. Clicking  in the object caption (if shown) produces the same result. This command is available only if minimizing is allowed in the object's **Properties** dialog on the *Caption (page 496)* page.
- Maximize** Enlarges the object to fill the screen. Clicking  in the object caption (if shown) produces the same result. This command is available only if maximizing is allowed in the object's **Properties** dialog on the *Caption (page 496)* page.
- 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 QlikView help.
- Remove** Removes the sheet object.

## 41.2 Line/Arrow Object Properties: General

This property page is opened by a right-click on a line/arrow object, or by selecting **Properties** in the **Object** menu when a line/arrow object is active.

In the **Orientation** group you determine the general direction of the line/arrow object by selecting one of the alternatives: **Horizontal**, **Vertical**, **Diagonal(Climbing)** or **Diagonal(Falling)**.

In the **Style** group you can alter the style of the line/arrow object to be used.



*Line/Arrow Object Properties, General*

- Line Style** Choose between a solid line and several types of dashed and/or dotted ones.
- Line Weight** Determines the thickness 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).



**Arrow Style** Several choices of style and orientation of the arrow head(s) are available here.

In the **Color** group you can set the color of the line/arrow object by clicking the colored button which opens the *Color Area (page 428)* dialog.

In the **Background** group you may define a background of the line/arrow object.

**Color** Choose this alternative if you want the object to be displayed with a colored background. The color can be defined as a solid color or it may be dynamically calculated from an expression via the **Color Area** dialog that opens when you click the colored button to the right of the radio button.

**Image** If you select this alternative, you will have to import a picture by clicking the **Select Image** button.

**Transparency** This alternative sets the degree of transparency of the object background. At 0% the background will be completely opaque. 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 **Object ID** is used for *Internal Macro Interpreter (page 893)* purposes. Every sheet object is assigned a unique ID, which for line/arrow objects start with LA01. Linked sheet objects share the same object ID. You may edit this ID number later on.

**Alternate State** Choose one of the available states in the list. The following Alternate States are always available.

**Inherited**

The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.

**Default state**

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

## 41.3 Actions

In the Actions tab 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 *Actions (page 552)*.

## 41.4 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

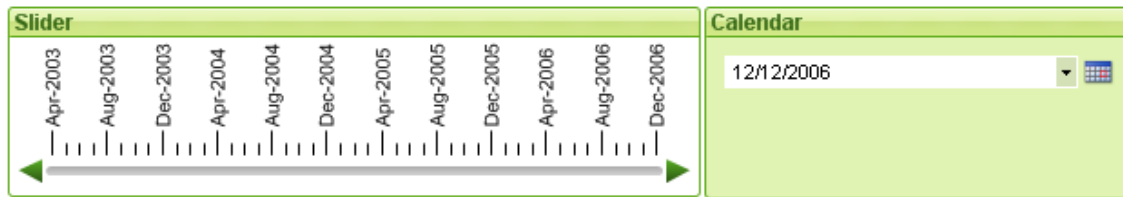
## 41.5 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



## 42 Slider/Calendar Object



Slider/Calendar objects provide alternative means for selecting field values in QlikView. They can also be used for entering values in QlikView *Variable* (page 173). As its name suggests, the slider/calendar object has two completely different modes. As different as they may look, they still work very much the same way behind the visual user interface.

Right-click on a slider/calendar object and the *Slider/Calendar Object: Object Menu* (page 571) will be displayed. It can also be accessed from the **Object** menu, when the slider/calendar object is the active object.

### 42.1 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 the figure below (page 575) 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 drop-down box with a calendar icon to the right. When clicking on the calendar icon it will expand to a calendar control. 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 variable.

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.

### 42.2 Slider/Calendar Object: Object Menu

By clicking once with the right mouse button while pointing on a slider/calendar object, a float menu appears. This menu can also be found under **Object** in the Main menu when a slider/calendar object is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Slider/Calendar Object Properties</b> dialog where <i>Slider/Calendar Object Properties: General (page 573)</i> , <i>Slider/Calendar Object Properties: Presentation (Slider Mode) (page 575)</i> , <i>Slider/Calendar Object Properties: Presentation (Calendar Mode) (page 576)</i> , <i>List Box Properties: Sort (page 486)</i> , <i>Properties: Number (page 490)</i> , <i>Font (page 491)</i> , <i>Layout (page 492)</i> , and <i>Caption (page 496)</i> parameters can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments (page 472)</i> for further information.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Image</b> 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 <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.

<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 42.3 Slider/Calendar Object Properties: General

The **Slider/Calendar Object Properties: General** page is opened by a right-click on a slider/calendar object and choosing the **Properties** command in the float menu. Here you can make general settings for the data to be manipulated by the slider/calendar object. It is also possible to reach this information via the dialog **New Slider/Calendar Object**.

Under **Input Style** select the basic mode of the slider/calendar object. Use the drop-down to select whether the object should be represented by a **Slider** control or a **Calendar** input control.

The **Data** group is where it is defined which data object the slider/calendar is connected to. A slider/calendar can be connected to either a field or to one or a pair of variables.

*Slider/Calendar Object Properties, General*

**Field** Select this radio button if 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 (page 757)* dialog for the expression chosen in the drop-down list.

**Variable(s)** Check the radio button and select the desired variable(s) in the drop-down list(s). The second variable is only available if you select **Multi Value** under **Mode** (see below).

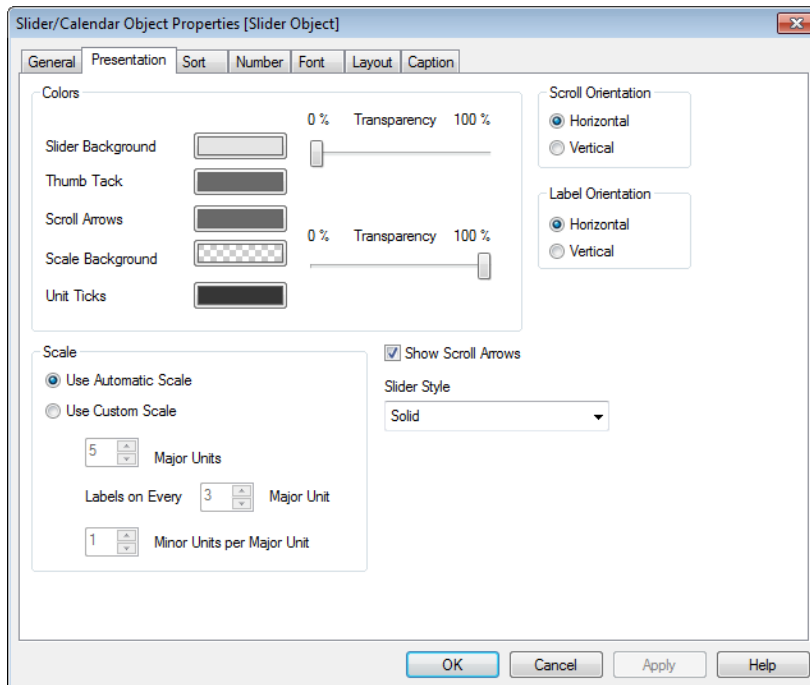
In the **Mode** group specify if the slider/calendar should be used for selecting a **Single Value** or a range of values - **Multi Value**.

In the **Value Mode** group you specify if the slider/calendar should select **Discrete** values (unavailable for **Multi Value** mode) or define a **Continuous/Numeric** range of values (unavailable for **Single Value** mode).

with **Field** data). This group is not available in **Calendar** mode. In **Calendar** mode discrete values are always assumed.

<b>Min Value</b>	Sets the minimum value for the slider/calendar in <b>Continuous/Numeric</b> value mode.
<b>Max Value</b>	Sets the maximum value for the slider/calendar in <b>Continuous/Numeric</b> value mode.
<b>Static Step</b>	Check the box and enter a value in the edit box to specify a static step for slider/calendar values in the <b>Continuous/Numeric</b> value mode.
<b>Value 1</b>	Shows current value for the slider in <b>Continuous/Numeric</b> value mode if the slider/calendar is <b>Single Value</b> . In <b>Multi Value</b> mode the low value is shown.
<b>Value 2</b>	Shows current high value for the slider/calendar in <b>Continuous/Numeric</b> value mode. Only relevant in <b>Multi Value</b> mode.
<b>Override Locked Field</b>	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.
<b>Fixed Range</b>	When this alternative is checked, the user will not be able to widen or narrow the range by dragging the edges of the slider thumb-tack.
<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .
<b>Object ID</b>	This is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for slider objects start with SL01. Linked sheet objects share the same object ID. It is possible to edit this ID number later on.
<b>Print Settings</b>	This button takes you to the <b>Print Settings</b> dialog where margins and header/footer format can be defined. The <b>Print Settings</b> dialog holds two pages, <i>Print: Layout (page 92)</i> and <i>Print: Header/Footer (page 93)</i> , which are identical to the last two pages of the <b>Print</b> dialog.

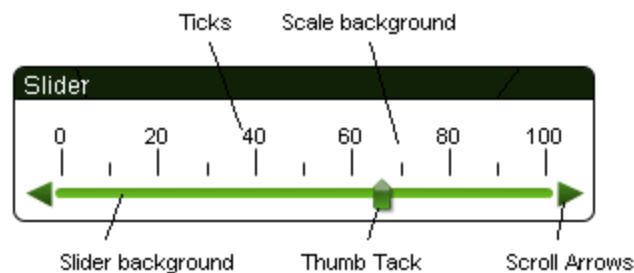
## 42.4 Slider/Calendar Object Properties: Presentation (Slider Mode)



Slider Object Properties, Presentation

The **Slider/Calendar Object Properties: Presentation** tab is opened by a right-click on a **Slider/Calendar** object and choosing the **Properties** command from the float menu. This is where the visual properties of the slider/calendar object can be configured.

The **Colors** group lets you define colors for the different parts of the slider/calendar, as shown in this figure:



**Slider Background** and **Scale Background** color can be defined as a solid color or a gradient via the *Color Area* (page 428) dialog that opens when you click the corresponding 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.

**Thumb Tack**, **Scroll Arrows** and **Unit Ticks** color can be defined as a solid color or a gradient via the **Color Area** dialog that opens when you click the corresponding button.

In the **Scroll Orientation** group the slider can be displayed in the **Horizontal** or **Vertical** direction.

In the **Label Orientation** you can determine whether scale text should be oriented in **Horizontal** or **Vertical** direction.

In the **Scale** group you can define a slider scale. Select **Use Automatic Scale** for the scale to be automatically generated based on slider size and underlying values. The alternative **Use Custom Scale** lets you specify a number of scale settings:

- Major Units**                      The number defines the number of major units on the slider scale.
- Labels 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.

Scroll arrows can be visible or hidden with the alternative **Show Scroll Arrows**.

**Slider Style** can be set as **Solid** or **Shimmer**.

---

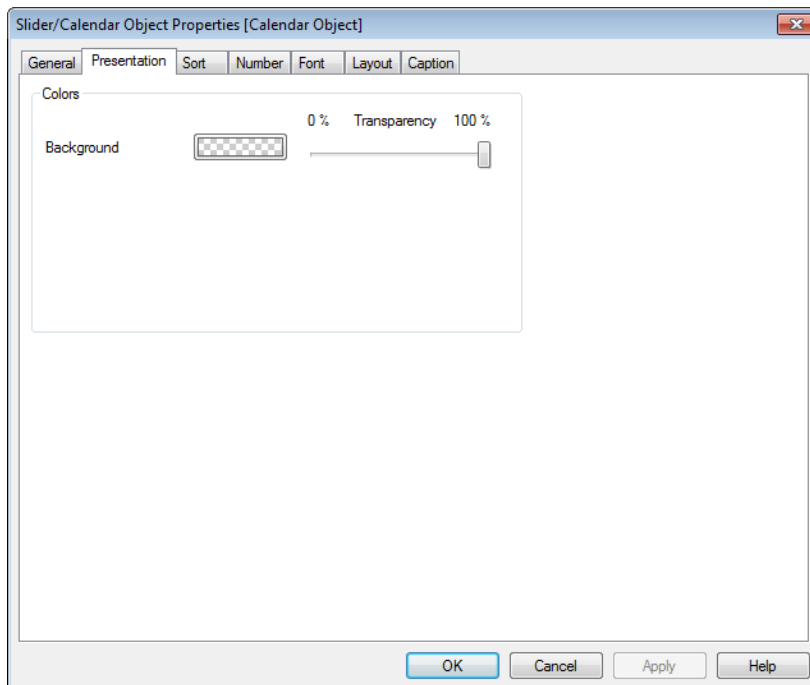
**Note!**

There are limitations to the automatic scale and vertical label orientation functions in AJAX/WebView mode.

If it is possible, it is recommended that you **Use Custom Scale** to define the scale settings.

---

## 42.5 Slider/Calendar Object Properties: Presentation (Calendar Mode)



*Calendar Object Properties, Presentation*

The **Slider/Calendar Object Properties: Presentation** tab is opened by a right-click on a **Slider/Calendar** object and choosing the **Properties** command from the float menu. This is where the visual properties of the slider/calendar object can be configured.

The **Colors** group lets you define the color of the calendar control background. The **Background** 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 calendar control 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.

## 42.6 Sort

On the **Sort** tab the sort order can be set for the slider data. This dialog is very similar to its counterpart under Document Properties (see *Document Properties: Sort (page 447)*).

## 42.7 Number

On the **Number** tab the number format can be set for the slider scale. This dialog is very similar to its counterpart under **Document Properties** (see *Document Properties: Number (page 449)*).

## 42.8 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 42.9 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 42.10 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



## 43 Bookmark Object

The bookmark object is a sheet object that is used for displaying bookmarks for selections. Depending on its configuration it can also be used for adding new bookmarks or deleting old ones. The bookmark object basically offers the options of the *Bookmarks Menu* (page 55).

By right-clicking the object, the *Bookmark Object: Object Menu* (page 579) will be displayed. It can also be accessed from the **Object** menu, when the bookmark object is the active object.




### 43.1 Bookmark Objects in Alternate States

Bookmark objects can be placed in an alternate state. However, bookmarks that are recalled and created through the object will apply to all states. There is no mechanism to act upon specific states with bookmark objects. In short, the state of a bookmark object is ignored.

### 43.2 Bookmark Object: Object Menu

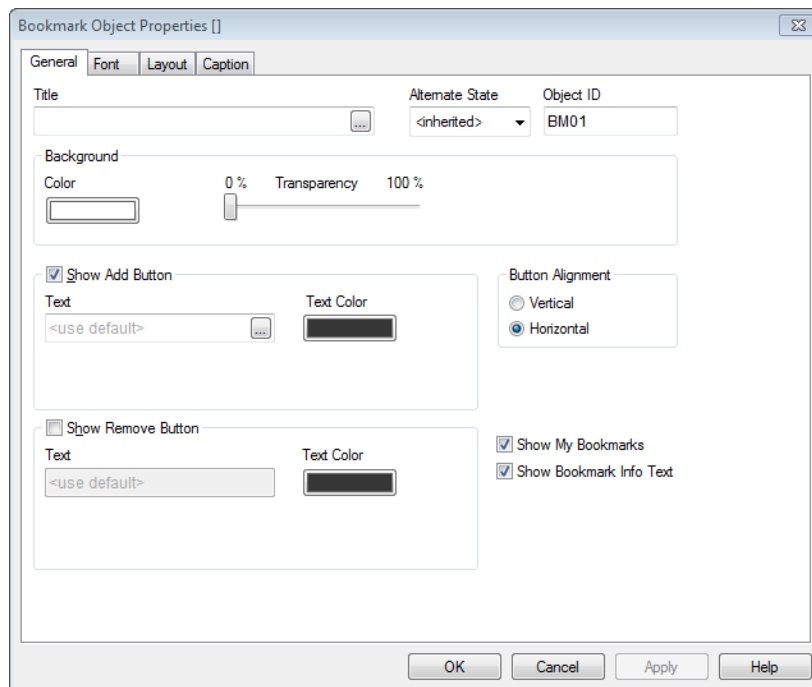
By clicking once with the right mouse button while pointing on a bookmark object, a float menu appears. This menu can also be found under **Object** in the Main menu when a bookmark object is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <i>Bookmark Object Properties: General</i> (page 580) dialog where you can set the parameters defining the bookmark object.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Order</b>	This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127. <b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet. <b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet. <b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127. <b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.
<b>Add Bookmark</b>	Opens the <b>Create Bookmark</b> dialog where the bookmark name can be edited.
<b>Replace Bookmark</b>	Opens a cascade menu with the ten most recently used bookmarks currently defined in the document. If you choose one of these, the contents of that bookmark will be replaced with the current state of selections and variable values.
<b>Remove Bookmark</b>	Opens a cascade menu with the ten most recently used bookmarks currently defined in the document. Selecting one of these removes it from the document.
<b>Import Bookmarks...</b>	After browsing for and selecting a previously saved bookmark (.qbm) file, the <i>Import Bookmarks</i> (page 122) dialog will open to let you import bookmarks.

<b>Export Bookmarks...</b>	Opens the <i>Export Bookmark(s)</i> (page 121) dialog where you can export selected bookmarks to a QlikView bookmark (.qbm) file.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Image</b> 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 <b>User Preferences</b> dialog, <b>Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption</i> (page 496) page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption</i> (page 496) page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the sheet object.

## 43.3 Bookmark Object Properties: General

The **Bookmark Object Properties: General** tab is opened by a right-click on a bookmark object and choosing the **Properties** command in the float menu. Here it is possible to set general parameters for the bookmark object.



*Bookmark Object Properties, General*

- Title** The text to appear in the bookmark object caption area. The title can be defined as a *Calculated Formula* (page 871) for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.
- Alternate State** Choose one of the available states in the list. The following Alternate States are always available.
- Inherited**  
The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.
- Default state**  
This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.
- Object ID** This is used for *Internal Macro Interpreter* (page 893) purposes. Every sheet object is assigned a unique ID, which for bookmark objects start with BM01. Linked objects share the same ID. It is possible to edit this ID number later on.
- Background**
- Color**  
This can be defined as a solid color or a gradient via the *Color Area* (page 428) dialog that opens when you click the 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. At 100% the background will be completely transparent.

- Show Add Button** Check this alternative to display an **Add Bookmark** button in the bookmark object. This option is selected by default.  
**Text**  
Here you may type a text to be shown on the **Add Bookmark** button. This can be defined as a *Calculated Formula (page 871)* for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas. 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.
- Button Alignment** Here you decide how the **Add** and **Remove** buttons are to be positioned in relation to each other, when both are shown: Choose between **Horizontal** (side by side) or **Vertical** (stacked) orientation.
- Show Remove Button** Mark this checkbox if a **Remove Bookmark** button is to appear in the bookmark object. This option is deselected by default.  
**Text**  
Here you may type a text to be shown on the **Remove Bookmark** button. This can be defined as a calculated label expression for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas. 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.
- Show My Bookmarks** Check this option 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** Check this option in order to display the text entered under **Edit Info**.

## 43.4 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 43.5 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 43.6 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 44 Search Object

Search objects can be used for searching for information anywhere in the document.

To create a new **Search Object** select **Layout - New Sheet Object - Search Object** in the Main Menu, right-click in the sheet area and select **New Sheet Object - Search Object** or click on the **Create Search Object** tool in the toolbar (if the tool has been activated).

By a right-click on a search object the *Search Object: Object Menu (page 583)* will be displayed. It can also be accessed from the **Object** menu, when the search object is the active object.

### 44.1 Search Object: Object Menu

By clicking once with the right mouse button while pointing on a search object, a float menu appears. This menu can also be found under **Object** in the Main menu when a search object is active. The menu contains the following commands:

<b>Properties...</b>	Opens the <i>Search Object Properties: General (page 584)</i> dialog where you can set the parameters defining the search object.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments (page 472)</i> for further information.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Image</b> 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 <b>User Preferences</b> dialog, <b>Export</b> page.</p> <p><b>Object</b> 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.</p>

**Linked Objects**

Opens a menu with the following commands for linked objects.


**Adjust Position of Linked Objects**

All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.


**Unlink This Object/Unlink Objects**

This destroys the link between the objects, making them different objects with different object IDs.


**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 object's **Properties** dialog on the *Caption (page 496)* page.

**Maximize**

Enlarges the object 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 object's **Properties** dialog, on the *Caption (page 496)* page.

**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**

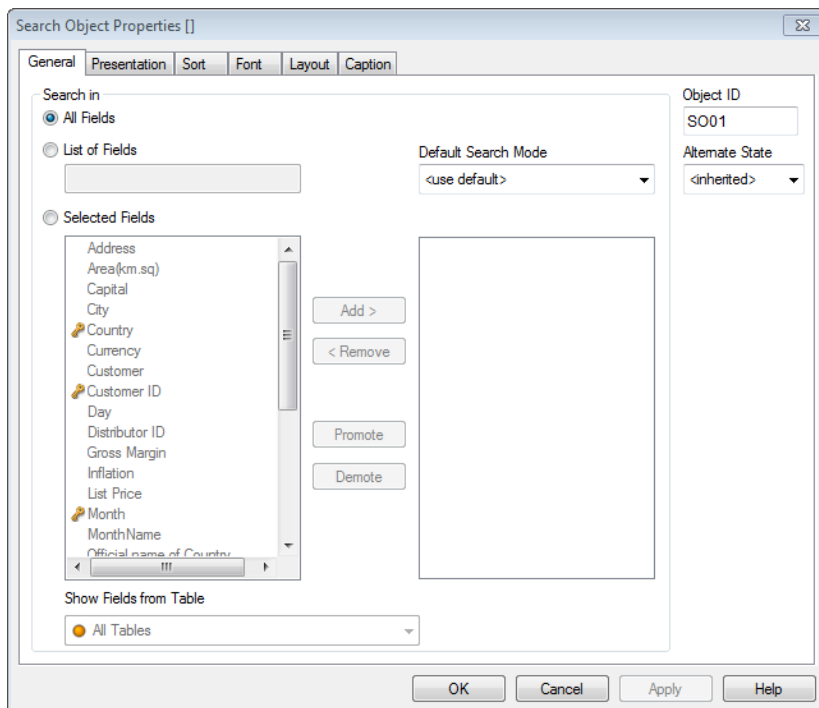
Opens QlikView help.

**Remove**

Removes the sheet object.

## 44.2 Search Object Properties: General

On the **General** tab you can configure the search options.



*Search Object Properties, General*

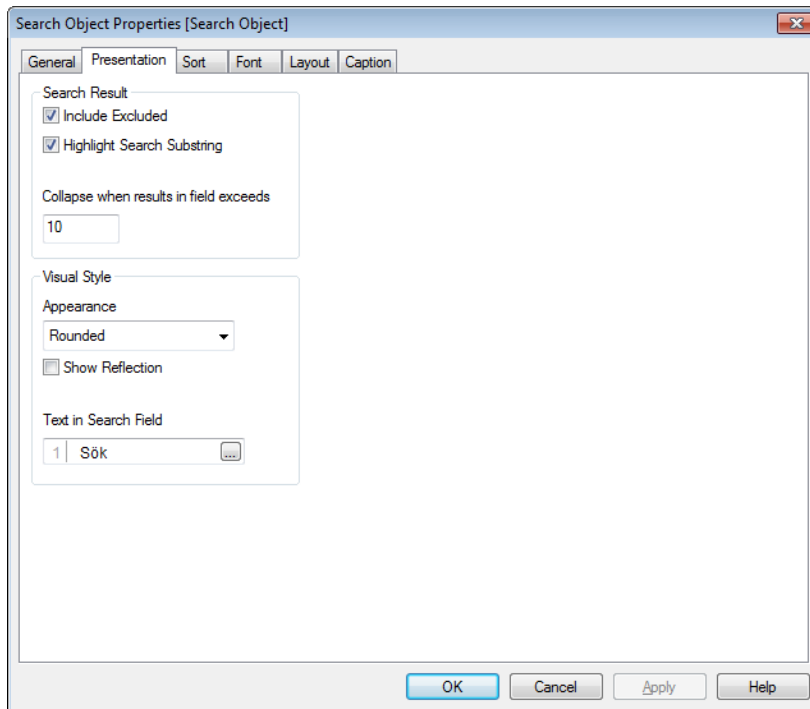


---

<b>All Fields</b>	Mark this radio button to allow the search object to search all fields in the document.
<b>List of Fields</b>	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 <b>calculated formula</b> for dynamic update.
<b>Selected Fields</b>	Mark this radio button and choose what fields should be searchable.
<b>Show Fields from Table</b>	This drop-down menu shows the value <b>All Tables</b> which contains all fields of the document. By selecting a specific table name in the drop-down menu you may limit the <b>Selected Fields</b> list to only fields from that internal table. Finally, you may select the value <b>All Tables (Qualified)</b> 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.
<b>Default Search Mode</b>	<p>Specifies the initial default search mode to be used in text searches. The mode can always be changed on the fly by typing * or ~ as part of the search string. The following alternatives are available:</p> <p><b>&lt;use default&gt;</b> The default as specified under <i>User Preferences (page 69)</i> will apply.</p> <p><b>Use Wildcard Search</b> The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search.</p> <p><b>Use Fuzzy Search</b> The initial search string will be a tilde (~) to denote a fuzzy search.</p> <p><b>Use Normal Search</b> No additional characters will be added to the search string. Without wildcards, a Normal Search will be made.</p> <p><b>Use Associated Search</b> The search box will open using the Associated Search.</p>
<b>Object ID</b>	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. Linked sheet objects share the same object ID. 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.
<b>Alternate States</b>	<p>Choose one of the available states in the list. The following Alternate States are always available.</p> <p><b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.</p> <p><b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b>.</p>

---

## 44.3 Search Object Properties: Presentation



*Search Object Properties, Presentation*

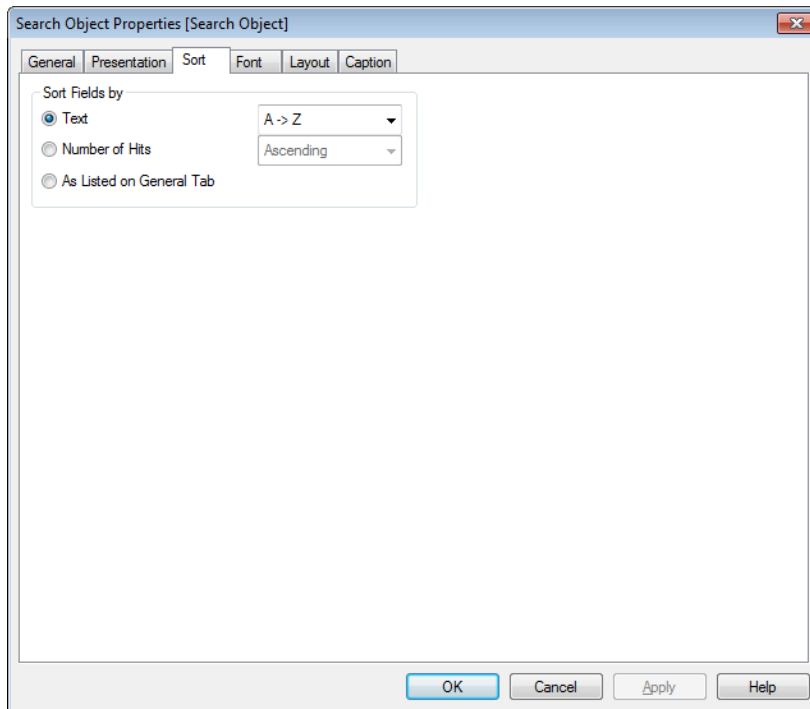
### Search Result

- 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 result 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* (page 757) is opened for easier editing of long formulas.

## 44.4 Search Object Properties: Sort



*Search Object Properties, Sort*

In **Sort Fields by** group the sort order of the fields in the search object can be set.

<b>Text</b>	Sorts the fields containing search hits in alphabetical order.
<b>Number of Hits</b>	Sorts the fields containing search hits according to the number of hits in each field.
<b>As Listed on General Tab</b>	Sorts the fields containing search hits according to the order in which the fields are listed on the <b>General</b> tab.

## 44.5 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font* (page 491).

## 44.6 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout* (page 492).

## 44.7 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 45 Container

Author	Title
Adams, Douglas	Hitchhikers Guide to the Galaxy, The
Adams, Douglas	Restaurant at the End of the Universe, ...
Adams, Douglas	So Long and Thanks for the Fish
Adams, Douglas	Titanic
Benson, Amber	Death's Daughter
Briggs, Patricia	Blood Bound
Briggs, Patricia	Bone Crossed
Briggs, Patricia	Iron Kissed
Briggs, Patricia	Moon Called
Eddings, David	Pawn of Prophecy
Eddings, David	Redemption of Althalus, The
Eddings, David	Seeress of Kell, The
Eddings, David	Wizard's Endgame
Evanovich, Janet	Four to Score
Evanovich, Janet	One for the Money
Evanovich, Janet	Ten Big Ones

The container is an object that contains other objects. The container can contain all other sheet objects. The objects are grouped together and have common settings for font, layout and caption.




By a right-click on the container the *Container: Object Menu (page 589)* will be displayed. It can also be accessed from the **Object** menu, when the container is the active object.

### 45.1 Container: Object Menu

The Object menu is found as the Object menu when an object is active. It can also be opened as a context menu by right-clicking on the object.

The following commands are available:

- Properties...** Opens the **Properties** dialog. This command can also be invoked by the following keyboard shortcut: Alt+Enter.
- Notes** Allows creating and sharing notes about the current object. See *Notes and Comments (page 472)* for further information.
- Order** This cascade menu is only available when the **Design Grid** command of the *View Menu (page 52)* is activated or when the *Always Show Design Menu Items (page 78)* checkbox is marked. 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. The maximum value is 127.
  - Send Backward** Decreases the layout layer of the sheet object by one. The minimum value is -128.

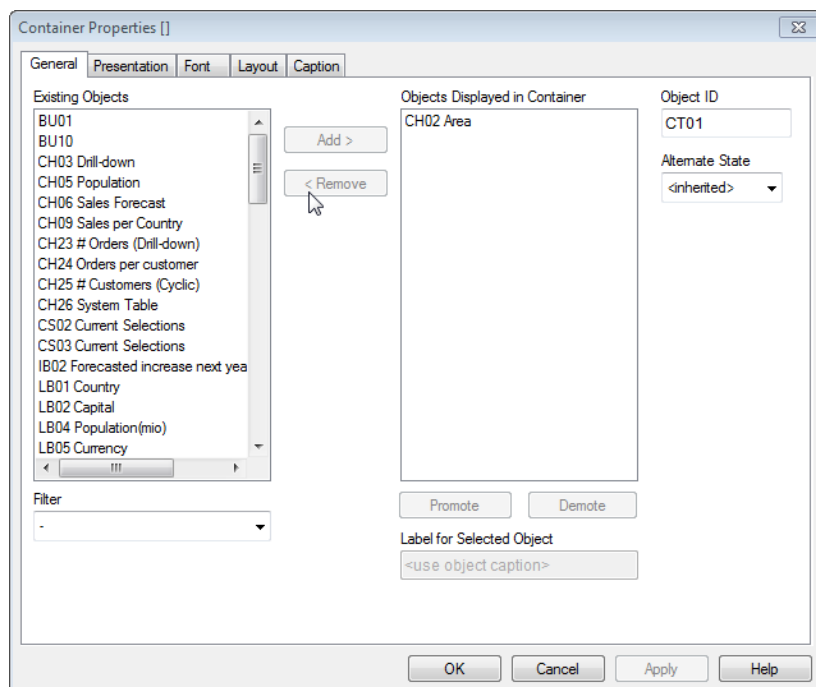
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Image</b> Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences</b> dialog, <b>Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	<p>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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Maximize</b>	<p>Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Restore</b>	<p>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.</p>
<b>Help</b>	<p>Opens QlikView help.</p>
<b>Remove</b>	<p>Removes the sheet object.</p>

## 45.2 Container Properties: General

The **Container Object: General** tab is opened by right-clicking on a container and choosing **Properties**. If the **Properties** command is dimmed you probably don't have the privileges needed to perform property changes.

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.

In the **Title** window the container can be given a name which will be displayed in the window caption. The title can also be defined as a *Calculated Formula (page 871)* for dynamic update of the label text. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.



*Container Properties, General*

- Existing Objects** A list of the available objects in the document. Select the items to be used/removed by clicking them. Use the **Add >** or the **< Remove** button to move them to the desired column.
- Filter** With this setting it is possible to filter the list of **Existing Objects** by, for example, object type.
- Objects Displayed in Container** A list of the objects already in the container object.
- Promote** Moves the selected object one step up.
- Demote** Moves the selected object one step down.
- Object ID** The unique ID of the current container. Upon creation every sheet object is assigned a unique ID for control via Automation. Linked objects share the same object ID. The ID consists of a combination of letters defining the type of object, as well as of a number. The first container in a document will be assigned the ID CT01. An object ID may be changed to any other string that is currently not being used for any other sheet object, sheet or bookmark in the document.

**Alternate State**

Choose one of the available states in the list. The following Alternate States are always available.

**Inherited**

The sheets and sheet objects are always in the **inherited** state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen.

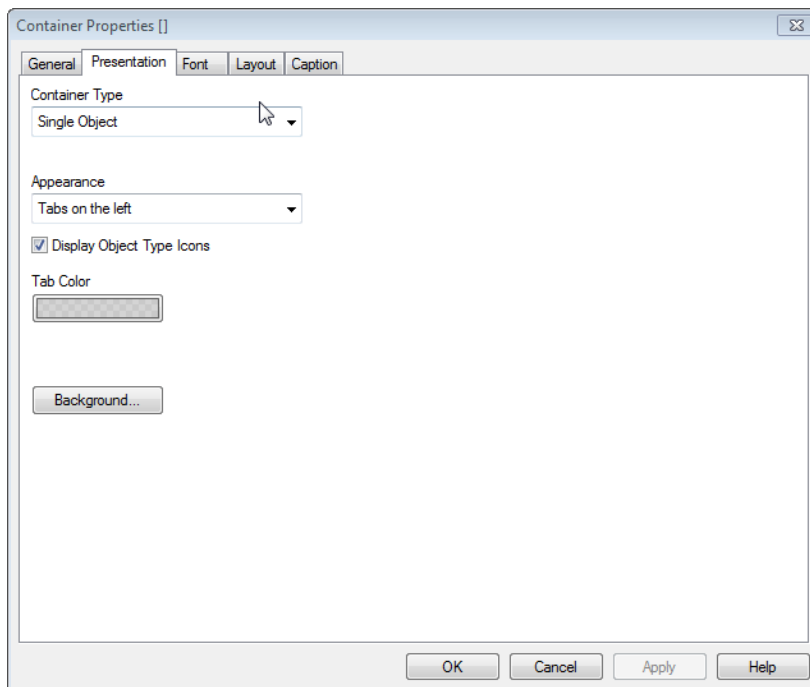
**Default state**

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the **default state**.

**Label for Selected Object**

An alternative name for the displayed object can be entered here. The label may be defined as a *Calculated Formula* (page 871) for dynamic update. By clicking the ... button the full **Edit Expression** dialog is opened for easier editing of long formulas.

## 45.3 Container Properties: Presentation



*Container Properties, Presentation in Single Object mode*

Here you can set properties concerning the appearance of the container.

**Container Type**

Select container type from the drop-down list.



<b>Appearance</b>	Here you set how the different objects in the container should be displayed. Only available in Single Object mode. <b>Tabs at top</b> All objects in the container are displayed as tabs at the top of the container. <b>Tabs on the left</b> All objects in the container are displayed as tabs on the left hand side of the container. <b>Tabs to the right</b> All objects in the container are displayed as tabs on the right hand side of the container. <b>Tabs at bottom</b> All objects in the container are displayed as tabs in the lower part of the container. <b>Dropdown at top</b> All objects in the container are displayed in a drop-down menu at the top of the container.
<b>Display Object Type Icons</b>	Enable this setting to have icons symbolizing the different objects in the tabs. Only available in Single Object mode.
<b>Tab Color</b>	Specifies the color of the tabs. The color can be defined as a solid color or a gradient via the <b>Color Area</b> dialog that opens when clicking the button. Only available in Single Object mode.
<b>Columns</b>	Select number of columns. Only available in Grid mode.
<b>Rows</b>	Select number of rows. Only available in Grid mode.
<b>Spacing</b>	Select the size of the spacing. Only available in Grid mode.
<b>Background...</b>	Opens the <i>Background Settings (page 489)</i> dialog.

## 45.4 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 45.5 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 45.6 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



## 46 Custom Object

The custom object is a sheet object specifically intended to carry custom defined OCX replacement controls. The *Custom Object: Object Menu* (page 595) can be accessed from the **Object** menu, when the custom object is the active object.

Replacement controls are windowless OCX controls programmed by QlikTech, yourself or third party vendors, which comply with the specifications for QlikView replacement OCX controls. The OCX control will be drawn in a rectangle on the sheet outlined by the underlying custom object. Communication between the OCX control and the QlikView document is maintained via the *Internal Macro Interpreter* (page 893).

The QlikView installation comes with a few sample OCX replacement controls. These are provided as is and with full VB source code. Use them as they are or modify the source code to better suit your needs. QlikTech does not offer support on these sample controls.

### 46.1 Custom Object: Object Menu

By clicking once with the right mouse button while pointing on a custom object, a float menu appears. This menu can also be found under **Object** in the Main menu when a custom object is active. The menu contains the following commands:

<b>Properties for Replacement Control</b>	Opens the <b>Properties</b> dialog for the OCX replacement control used. This dialog is provided with the OCX replacement control and may be absent.
<b>Properties...</b>	Opens the <i>Custom Object Properties: General</i> (page 596) dialog.
<b>Order</b>	This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127. <ul style="list-style-type: none"> <li><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</li> <li><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</li> <li><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</li> <li><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</li> </ul>
<b>Copy to Clipboard</b>	This menu contains the various copying options for the sheet object. <ul style="list-style-type: none"> <li><b>Image</b> Copies a bitmap picture of the custom object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences</b> dialog, <b>Export</b> page.</li> <li><b>Object</b> 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.</li> </ul>

**Linked Objects**

Opens a menu with the following commands for linked objects.


**Adjust Position of Linked Objects**

All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.


**Unlink This Object/Unlink Objects**

This destroys the link between the objects, making them different objects with different object IDs.


**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 object's **Properties** dialog on the *Caption (page 496)* page.

**Maximize**

Enlarges the object 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 object's **Properties** dialog on the *Caption (page 496)* page.

**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**

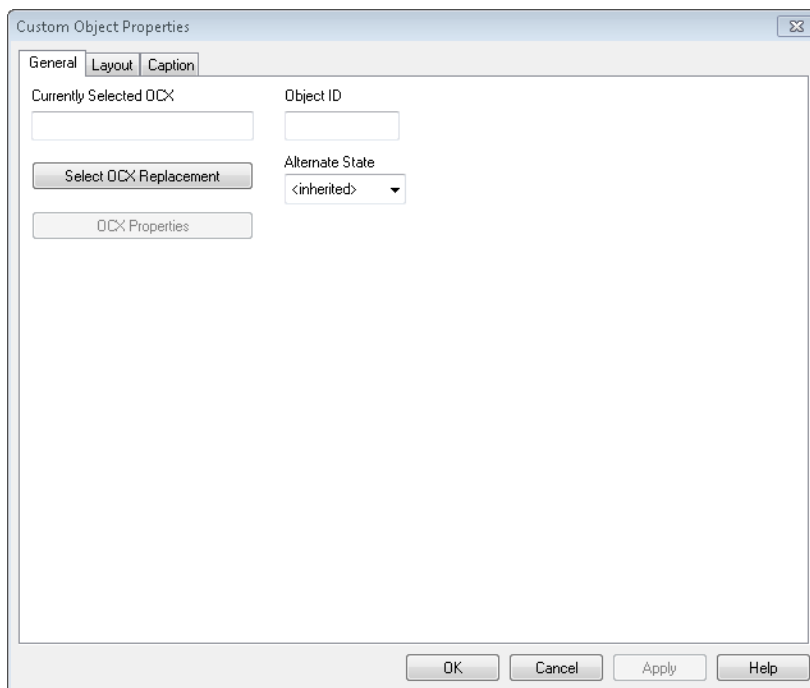
Opens QlikView help.

**Remove**

Removes the sheet object.

## 46.2 Custom Object Properties: General

The **Custom Object Properties: General** page is opened by a right-click on a custom object and choosing the **Properties** command in the float menu. This is where settings for title and configuration of the custom object can be made.



*Custom Object Properties, General*

<b>Currently Selected OCX</b>	The name of the currently selected OCX replacement.
<b>Object ID</b>	This is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for custom objects start with CO01. Linked objects share the same object ID. It is possible to edit this ID number later on.
<b>Select OCX Replacement</b>	Click this button to open the <b>Insert Object</b> dialog where you may select or change an OCX replacement control.
<b>OCX Properties</b>	Opens the <b>Properties</b> 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.
<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .

## 46.3 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 46.4 Caption

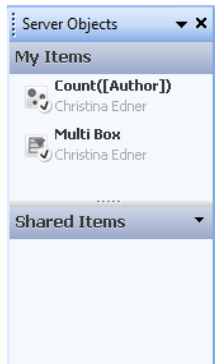
On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



## 47 Server Objects Pane

The **Server Objects Pane** offers an easy way for end-users to manage their own server sheet objects and access other users' shared objects.



*The Server Objects Pane*

### 47.1 Configuring the Pane

The **Server Objects Pane** can be toggled on and off via the **Server Objects** command on 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 **Server Objects Pane** can be left on when working with local documents (non server-based) but then has no function.

The **Server Objects 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 **Remove**. In order to show the object in the layout again, you must drag the object from the **Server Objects** pane.

When hovering over an object, a pop-up is shown containing information about the name of the object, the object type, owner and latest modification date.

### 47.2 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.

Those objects shown in the layout are marked by a check in the **My Objects** and **Shared Objects** areas.

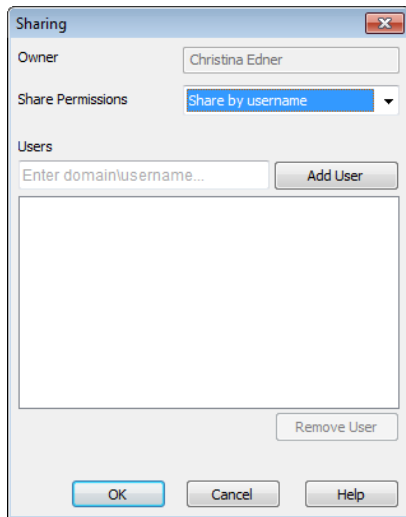
To use someone else's shared object you must drag it to the application window.

### 47.3 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 the object in the **My Objects** area and choose **Share with Everyone** or **Share Permissions...**. This opens the **Sharing** dialog.

## Sharing



*The Sharing dialog of the Server Objects 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 everybody**        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 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.



# 48 Layout Themes

## 48.1 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* (page 603), 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* (page 603) 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 for once one sheet object type which usually uses caption (list boxes, charts etc) and once for one sheet object type which normally goes without caption (text object, button etc).
4. Run the **Theme Maker Wizard** creating a new theme and insert properties from the first of the formatted entities listed above (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.

## 48.2 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 on it.
2. Open the **Properties** dialog for the sheet object from the context menu.
3. Go to the **Layout** page
4. Click on 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 lasso selection.

## Applying a Theme to a Sheet

Do as follows:

1. Activate the sheet by clicking on its tab.
2. Open the **Sheet Properties** dialog from the **Settings** menu.
3. Go to the **General** page
4. Click on 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 will now be applied. Additionally the theme will also be applied to all sheet objects on the sheet.

## Applying a Theme to a Whole 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 on 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.

## 48.3 Theme Maker Wizard

<b>Step 1 - Select Theme File</b>	<i>page 603</i>
<b>Step 2 - Source Selection</b>	<i>page 603</i>
<b>Step 3 - Select Specific Properties</b>	<i>page 604</i>
<b>Step 4 - Insertion of Properties in Theme</b>	<i>page 604</i>
<b>Step 5 - Save Theme</b>	<i>page 604</i>

Select **Theme Maker Wizard** command from the **Tools** menu to start the Theme Maker wizard.

The first time a start page outlining the purpose and the basic steps of the wizard is opened. To skip the start page in the future, mark the **Don't show this page again** check box. Click **Next** to continue.

### Step 1 - Select Theme File

Choose between creating a new theme from scratch, creating a new theme based on an existing one or modifying an existing theme.

- |                              |   |
|------------------------------|---|
| <b>New Theme</b>             | Choose this option to create a new theme.   |
| <b>Template</b>              | To base the 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 a <b>Browse...</b> option for browsing for theme files in other locations is found. |
| <b>Modify Existing Theme</b> | To modify an existing theme choose this option. Select a theme in the drop-down. The drop-down will list all existing themes in the default QlikView theme folder. At the bottom of the list a <b>Browse...</b> option for browsing for theme files in other locations is found.    |

Click **Next** to continue. The **Save As** dialog will appear when creating a new theme.

### Step 2 - Source Selection

When building a theme 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.

- |               |   |
|---------------|---|
| <b>Source</b> | Select the source object from the list in the drop-down box. 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 such properties that only exist in a given object type, e.g. charts. This type of properties can only be copied to other objects of the same type as the source object type.

**Caption & 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 wizard steps that follow.

Click **Next** to continue.

## Step 3 - Select Specific Properties

In step 3 more detailed selections of properties to be extracted from the source object and included in the theme can be made. 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 the items to include.

When modifying an existing theme, those items marked when entering this step are those currently included in the theme. By changing selections all previous settings in the theme will be overwritten.

When creating a new theme from scratch, those items marked when entering 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 Theme

In the fourth step in 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.

Those check boxes surrounded by green frames indicate sub-sections which 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

There are two options available for a theme to be saved as defaults for new documents or objects.

**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 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. Default theme can be set at all times from the **Document Properties: Presentation** page.

**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 newly created 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 theme for 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.



---

# Part 6 Charts

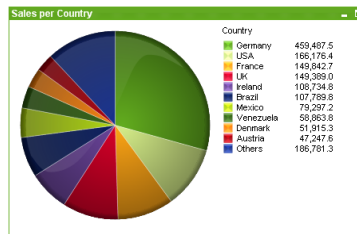
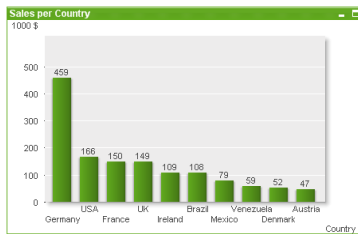
# 49 Introduction

Charts and tables are sheet objects that can show numbers very compactly. It is possible e.g. to 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.

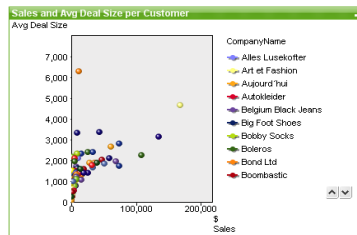
## The Different Chart Types Available

The chart types you can choose between are bar chart, pie chart, combo chart, scatter chart, line chart, radar chart, grid chart, gauge chart, block chart, funnel chart, pivot table, straight table, and mekko chart.



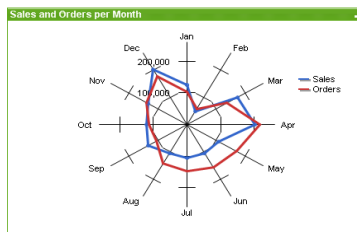
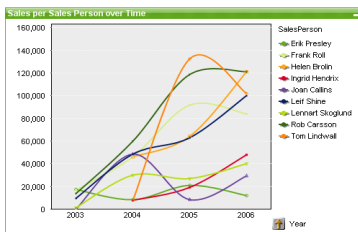
Bar chart

Pie chart



Combo chart

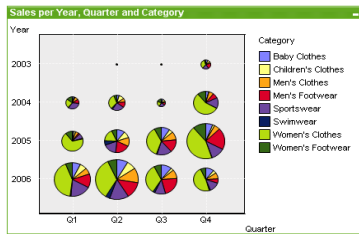
Scatter chart



Line chart

Radar chart

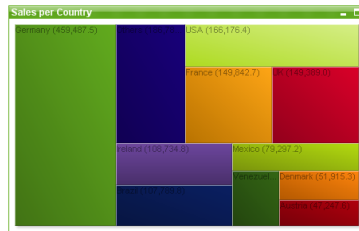




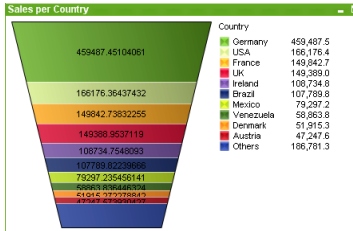
Grid chart



Gauge chart



Block chart



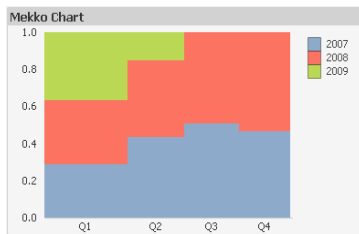
Funnel chart

Country	Salesman	Year	Sales
Australia	Rolf Wessenlund	2005	1,000
		2006	1,210
		<b>Total</b>	<b>2,240</b>
Azerbaijan			5,329
Bahrain			1,090
Bangladesh			4,240
Belarus	Charles Ingvar Jönsson	2006	26,065
		2008	1,210
		2009	3,159
		<b>Total</b>	<b>8,059</b>
Belgium	John Cleaves	2008	2,500
		2008	2,500
		<b>Total</b>	<b>2,500</b>
Belgium	Tony Cedholt	2009	4,249
		<b>Total</b>	<b>6,749</b>
<b>Total</b>			<b>17,358</b>

Pivot table

CategoryName	ProductName	Sales	Quantity
		<b>\$1,665,625.31</b>	<b>51952</b>
Men's Clothes	Altes Lusseletta	\$30,126.55	1057
Men's Clothes	Boat fit	\$3,524.57	1315
Men's Clothes	Desperado Jeans	\$18,240.68	706
Men's Clothes	Lenin Jeansshorts	\$14,800.64	628
Men's Clothes	MC2 Trousers	\$17,244.48	1067
Men's Clothes	O-Man Underwear	\$1,649.87	298
Men's Clothes	Rossi Bermuda Shorts	\$10,847.25	1397
Men's Clothes	Samba Soccer Socks	\$4,941.14	1175
Men's Clothes	US-Master Jeans	\$21,756.94	817
Women's Clothes	Chantrel Shirt	\$7,504.70	388
Women's Clothes	Halter Dress	\$361,896.85	981
Women's Clothes	Jack Flash Dress	\$42,635.00	722
Women's Clothes	Langoste Shirt	\$4,433.35	246
Women's Clothes	Le Baby Dress	\$47,571.88	623
Women's Clothes	Mirini Palas	\$10,472.71	104
Women's Clothes	Oksaba Skin Jackets	\$42,258.78	601
Women's Clothes	Oyaki Kimono	\$9,084.42	806

Straight table

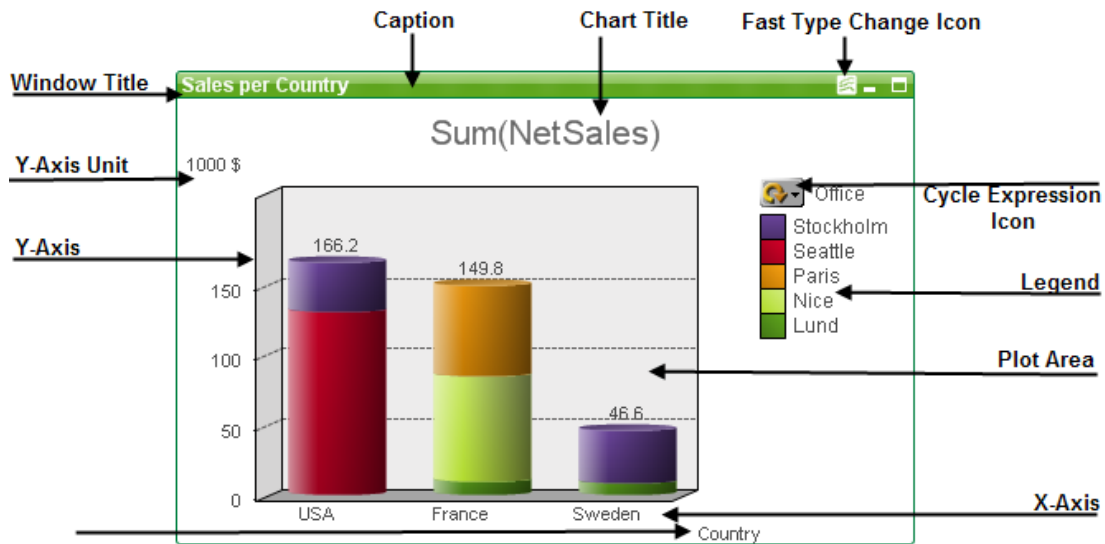


Mekko chart

QlikView charts can be divided into two major categories. The first, graph charts, consists of bar, line, combo, pie, scatter, radar, grid, block, mekko 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* (page 612) 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	
6580 Repairs and maintenance	Cash Flow aus B...	44.299	26.229	-69%	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	
<b>Total</b>		<b>2.051.337</b>	<b>3.031.724</b>	<b>32%</b>	<b>2.051.337</b>	<b>3.031.724</b>	
Cost of Sales		1.285.848	1.405.568	9%	1.285.848	1.405.568	
Other Inco...		-2.001.455	522.694	483%	-2.001.455	522.694	
Provision f...		14.789	42.048	65%	14.789	42.048	
Revenue		-3.374.727	-4.035.386	16%	-3.374.727	-4.035.386	
<b>Total</b>		<b>-2.024.208</b>	<b>866.648</b>	<b>309%</b>	<b>-2.024.208</b>	<b>866.648</b>	

The pivot table above shows some of the more common components of a QlikView table chart.

## 49.1 New Chart

A new chart can be created by clicking on the **Create Chart** tool in the toolbar, by a right-click on an empty sheet area and selecting **New Sheet Object, Chart** from **Object** menu or by selecting **New Sheet Object, Chart** from **Layout** menu. This will open a sequence of chart property pages.

In the page that appears first, name and type of graph and title (optional) is set, clicking the **Next** button opens the second page etc. As soon as sufficient information has been entered the **Next** and/or **Finish** buttons are enabled and the user can proceed to the next page in the sequence or finish it.

Once the chart is displayed on the sheet it can be modified by right clicking the graph and selecting **Properties** or by activating the graph (click on caption area) and selecting **Properties** in the **Object** menu.

## 49.2 Selections in Charts and Tables

The selection procedure in charts and tables has been designed to be as intuitive as possible. In most cases you will find it possible to make selections directly in the chart plot area by clicking individual values (bars etc.) or by "painting" over a larger selection.

If you find that you cannot make selections in a chart as indicated, the chart may be in **Detached** mode or **Read Only** mode (see *Chart Properties: General* (page 613)).

### Selections in 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, a green raster indicates the selection until the mouse button is released. The selection will be made for the dimension values used for calculating 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.

---

**Note!**

Selections made directly in line charts and bar charts (not combo charts) showing more than one dimension will primarily apply to just one dimension. Selections in line charts primarily apply to the second dimension so that painting over a line selects the entire line over all x-axis dimension values. In bar charts the opposite applies, i.e. selections primarily apply to the first dimension.

---

### Selections in Pie Charts

Selections can be made inside the plot area by clicking on a single pie slice or painting over several slices. When painting, a green raster indicates the selection until the mouse button is released. The selection will be made for the dimension values used for calculating the selected data point(s).

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

### Selections in Block Charts

Selections can be made inside the plot area by clicking a single block or by painting over several blocks. When clicking, selections are made with drill-down functionality so that the first click selects a single value in the 1st dimension. Clicking a second time (i.e. within the selection) selects a single value in the 2nd dimension and so on.

When painting, a green raster indicates the selection until the mouse button is released. The selection will be made for the dimension values used for calculating the selected data point(s). By painting across the boundaries of several values of the 1st dimension, all sub-values of the 2nd and 3rd dimensions belonging to these values are selected, not just the ones touched by the selection.

### Selections in Gauge Charts

Selections cannot be made in gauge charts.

### Selections in 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.

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

## Selections in 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.

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 for calculating the selected expression cell.

## Selections in Table Boxes

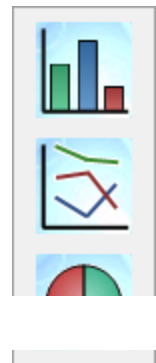
Selections can be made by clicking in any cell or by painting over an area covering one or more rows and one or more columns. The selected area is marked green until the mouse button is released.

## 49.3 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 **Chart Properties: General** page.

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.



## 49.4 Sizing and Moving Chart Components

Many of the individual chart components can be resized or moved to suit your preferences.

By pressing the Shift and Ctrl keys and keeping them depressed while a chart is active you will enter the chart layout edit mode. In edit mode, thin red rectangles will appear around those components of the chart that can be sized or moved. Use the mouse drag and drop technique to move things around.

The following components can be edited:

The **chart title** and **chart legend** can be both moved and resized. They can be docked to the top, bottom, left and right border of the chart and also be positioned free-floating anywhere within the chart.

Free-floating chart text can be moved to a position anywhere in the chart. The outlining rectangle can be resized to accommodate longer or multiline text.

The areas taken up by the **chart axes** and their **labels** can be resized.

**Cycle expression icons** and **fast chart type change icons** can be moved to a free-floating position anywhere within the chart.

The **plot area** itself cannot be resized or moved in edit mode, but will be drawn in the available space between the axes and docked legend and title.

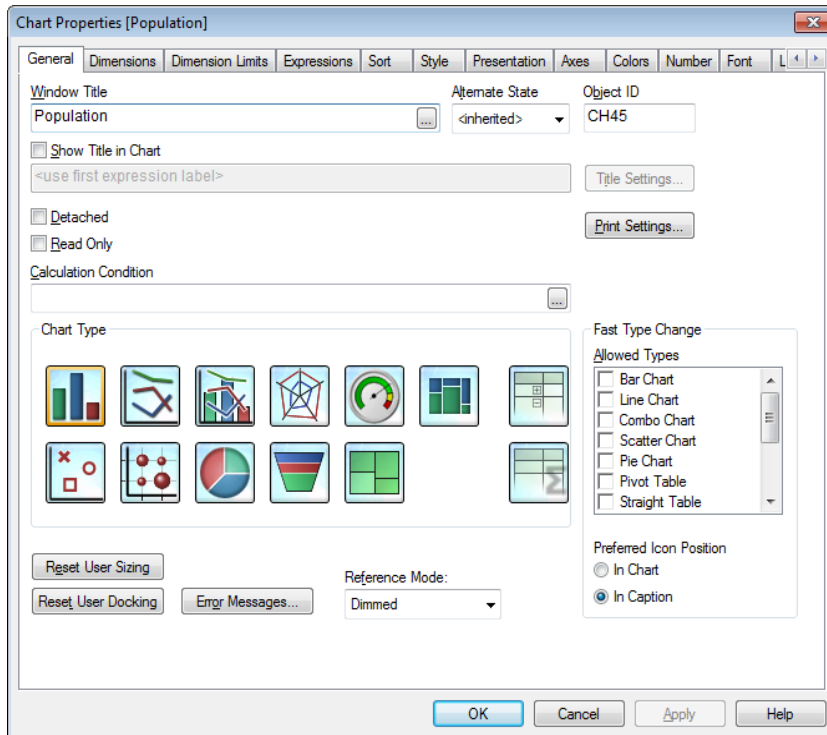
## 49.5 Chart Properties

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, that is 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 lack the privileges needed to perform property changes (see *Sheet Properties: Security (page 467)*).

The settings available in the **Chart Properties** dialog's different property 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.

## 49.6 Chart Properties: General



*Chart Properties, General*

On the **General** page you can set such properties as titles and chart type. It is the first page in the **Quick Chart Wizard** and in the **Chart Properties** dialog.

- Window Title** The title to be displayed in the window header. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*.
- Show Title in Chart** By default, the label of the first expression defined is set as chart title. Clear the check box if no chart title should be displayed. To display the original title, simply mark the check box. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*. The chart title is not displayed in pivot tables or straight tables.
- Title Settings** Define advanced settings for the chart title by clicking the **Title Settings** button.

<b>Print Settings</b>	Clicking the <b>Print Settings</b> button takes you to the <b>Print Settings</b> dialog where it is possible to define margins and header/footer format. The <b>Print Settings</b> dialog holds two pages, <i>Print: Layout (page 92)</i> and <i>Print: Header/Footer (page 93)</i> .
<b>Alternate State</b>	Choose one of the available states in the list. The following Alternate States are always available. <b>Inherited</b> The sheets and sheet objects are always in the <b>inherited</b> state unless overridden by the QlikView developer. This setting is inherited from the object on the level above, a chart in a sheet gets the same settings as the sheet if inherited is chosen. <b>Default state</b> This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the <b>default state</b> .
<b>Object ID</b>	The <b>Object ID</b> is used for <i>Internal Macro Interpreter (page 893)</i> purposes. Every sheet object is assigned a unique ID, which for charts start with CH01. Linked objects share the same object ID. It is possible to edit this ID number later on.
<b>Detached</b>	If enabled, the chart will be detached, that is, it will no longer be dynamically updated when selections are made.
<b>Read Only</b>	If enabled, the chart will be read only, that is, selections cannot be made by clicking or painting with the mouse in the chart.
<b>Calculation Condition</b>	Typing an expression in this text box sets a condition that needs to be fulfilled for the chart to be displayed. If the condition is not fulfilled, the text "Calculation condition unfulfilled" will be displayed in the chart. The value may be entered as a calculated formula. Click on the ... button to open the <i>Edit Expression Dialog (page 757)</i> .
<b>Chart Type</b>	The <b>Chart Type</b> group is where you select the basic layout of the chart. For more information on each chart type, see <i>Chart Types (page 615)</i> .
<b>Fast Type Change</b>	In this group you can enable an icon in the chart from which the user can change chart type without going through the chart properties dialog. <b>Allowed Types</b> In this list you select which chart types that should appear in the drop-down. Two or more types have to be chosen for fast type change to be enabled. <b>Preferred Icon Position</b> 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.
<b>Reset User Sizing</b>	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.
<b>Reset User Docking</b>	By pressing this button, all user docking of legend, title etc. in graphical charts will be reset.
<b>Error Messages</b>	Opens the <i>Custom Error Messages (page 875)</i> dialog.

**Reference Mode** Settings 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 charts.

## Chart Types

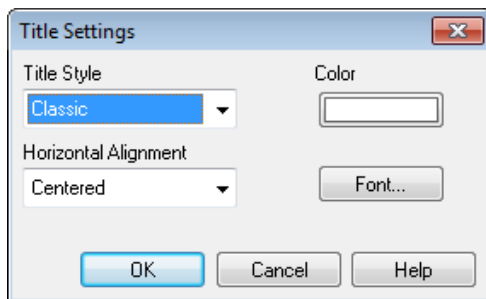
Charts are graphical representations of numerical data. Choose from the following chart types: **Bar, Line, Combo, Radar, Scatter, Grid, Pie, Funnel, Block and Gauge Chart** as well as **Pivot Table** and **Straight Table**. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General (page 613)* page.

A brief introduction to chart types:

<b>Bar Chart</b>	This is the most basic chart type. Each x-axis value corresponds to a bar. The bar height corresponds to its numerical y-axis value.
<b>Line Chart</b>	The line chart is essentially defined in the same way as the bar chart. Instead of using bars the data can be presented as lines between value points, as value points only or as both lines and value points (see <i>Chart Properties: Expressions (page 629)</i> ).
<b>Combo Chart</b>	The combo chart allows the combination of the features of the bar chart with those of the line chart. One expression will be displayed by lines and/or symbols, the other as bars.
<b>Radar Chart</b>	The radar chart is a variant of the line chart where the x-axis is plotted in a circle around the chart, resulting in a projection reminiscent of a radar screen or a spider's web.
<b>Scatter Chart</b>	The scatter chart plots data points representing combinations of expressions, iterated over one or several dimensions. Both axes are continuous, representing one expression each.
<b>Grid Chart</b>	The grid chart is a variant of the scatter chart that plots dimension values on the axes and uses an expression to determine the plot symbol. It can also show a third dimension in the form of small pie charts as plot symbols.
<b>Pie Chart</b>	Shows the relation between a single <i>Fields (page 171)</i> (primary dimension) and a single expression. A variant chart type is drawn when a secondary dimension is introduced. If more expressions than one are enabled in the <b>Chart properties: Expressions</b> page, the first in the expression list will be displayed. To switch expression use the <b>Promote/Demote</b> buttons in the <b>Expressions</b> property page.
<b>Block Chart</b>	The block chart shows the relation between expression values as blocks of varying area. It uses a single expression and up to three dimensions, with each dimension block further divided into sub-blocks. The total area of the block chart always equals 100% of the possible expression values. Sometimes a <i>Color Functions (page 375)</i> is used for creating a so called "heat chart".
<b>Funnel Chart</b>	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 without regards to data points.

<b>Gauge Chart</b>	Gauge charts are used to display the value of a single expression, lacking dimensions.
<b>Pivot Table</b>	The pivot table presents dimensions and expressions in table form. There is no formal limit to the number of dimensions or expressions possible. A pivot table can be defined without expressions, generating a tree view for navigating the dimension levels.
<b>Straight Table</b>	The straight table differs from the pivot table in that it can not display sub-totals and that the grouping of dimensions is shown in record form so that each row of the table contains field and expression values.

## Title Settings



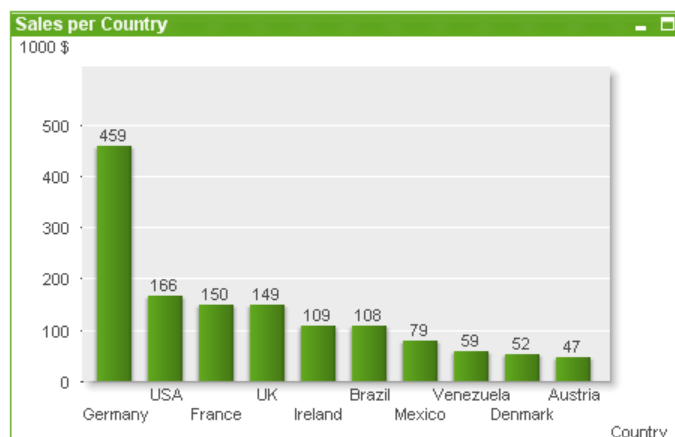
*The Title Settings dialog*

The layout of the chart title is controlled by the various settings of this dialog.

<b>Title Style</b>	Sets the basic style for the title.
<b>Background Color</b>	Sets the color of the title background. The color can be defined as a solid color or a gradient via the <b>Color Area</b> dialog that opens when clicking the button.
<b>Horizontal Alignment</b>	Specifies how the title text is positioned in relation to the plot area.
<b>Font</b>	Sets the font in the chart title. The standard Font dialog is opened when you click the button.



## 50 Bar Chart



The bar chart is the most basic chart type.

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

The quickest way to create a new bar chart is to select *Quick Chart Wizard* (page 767) from the **Tools** menu.

By right-clicking the object, the *Bar Chart: Object Menu* (page 617) will be displayed. It can also be accessed from the **Object** menu, when the bar chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

### 50.1 New Chart

A new chart can be created by clicking on the **Create Chart** tool in the toolbar, by a right-click on an empty sheet area and selecting **New Sheet Object, Chart** from **Object** menu or by selecting **New Sheet Object, Chart** from **Layout** menu. This will open a sequence of chart property pages.

In the page that appears first, name and type of graph and title (optional) is set, clicking the **Next** button opens the second page etc. As soon as sufficient information has been entered the **Next** and/or **Finish** buttons are enabled and the user can proceed to the next page in the sequence or finish it.

Once the chart is displayed on the sheet it can be modified by right clicking the graph and selecting **Properties** or by activating the graph (click on caption area) and selecting **Properties** in the **Object** menu.




### 50.2 Bar Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.

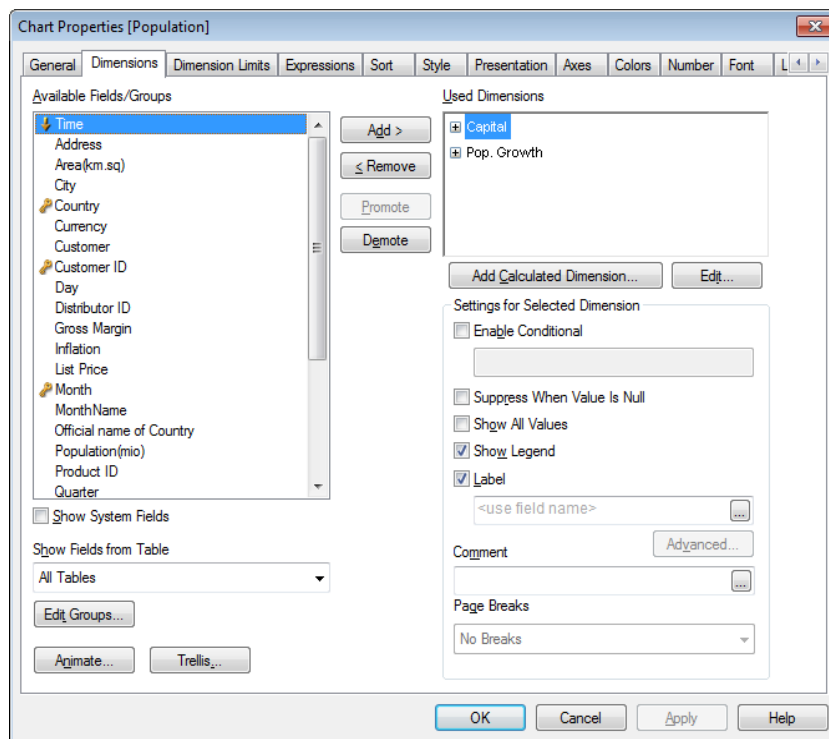
<b>Detach</b>	<p>The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached.</p> <p>By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.</p>
<b>Attach</b>	<p>Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.</p>
<b>Set Reference</b>	<p>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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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 a 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. The maximum number of objects that can be included when using the Set Reference option is 500.</p>
<b>Clear Reference</b>	<p>This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.</p>
<b>Clone</b>	<p>Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.</p>
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	<p>Clears all selections in the dimensions and expressions of the chart.</p>
<b>Print...</b>	<p>Opens the <i>Print: General</i> (page 91) dialog from which you can print the chart.</p>

<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer as default. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.
<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Values</b> Copies the values to the clipboard in the form of a table. <b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the sheet object.

## 50.3 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 50.4 Chart Properties: Dimensions



*Chart Properties, Dimensions*

The **Chart Properties: Dimensions** page is reached by right-clicking a chart and selecting **Properties** or by selecting **Properties** from the **Object** menu when the chart is active.

When you create a chart, you should first ask yourself two questions:

- 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** tab.
- 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** tab.

A chart can display one or more dimensions. The upper limit depends on the actual chart type, the complexity of the data and the available memory. Pie, line and scatter charts can display a maximum of two dimensions, bar, block and grid charts three. Radar charts can only display a single dimension, and gauge charts use no dimensions at all. A funnel chart typically has one to three dimensions. Further dimensions are ignored.

A chart dimension gets its values from a field which is specified on the **Chart Properties: Dimensions** page. Instead of being a single field a dimension can also consist of a group of fields (see **Edit Groups** below).

A dimension can be a single field, a group or an expression (calculated dimension). It can also be a synthetically created dimension, see *Synthetic Dimension Functions* (page 825).

Move fields back and forth by selecting (click, Ctrl-click) and using the **Add>** or **<Remove** buttons, or by double-clicking your selections directly.

Chart dimensions may also be calculated from an expression.

This property page is used for defining dimensions.

---

<b>Available Fields/Groups</b>	<p>Lists all fields/field groups that are available for use as dimensions (i.e. along the x-axis in a typical bar chart). Field groups will be preceded by a vertical arrow for <i>Hierarchic Groups (Drill-Down)</i> (page 873) or a curved arrow for <i>Non-Hierarchic Groups (Cyclic)</i> (page 874). Groups are defined in the <i>Document Properties: Groups</i> (page 442) page. Select the items to be used/removed by clicking them. Use the <b>Add</b> &gt; or the &lt; <b>Remove</b> button to move them to the desired column.</p> <p>The number of dimensions that can be displayed varies with different chart types, as outlined in the <i>Chart Properties: General</i> (page 613) page. All fields which appear in more than one internal 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.</p>
<b>Show System Fields</b>	<p>Checking this option will display the <i>System Fields</i> (page 379) in the <b>Available Fields/Groups</b> column.</p>
<b>Show Fields from Table</b>	<p>From here, you control what fields/groups appear in the <b>Available Fields/Groups</b> list. The drop-down list displays the alternative <b>All Tables</b> by default.</p> <p>The alternative <b>All Tables (Qualified)</b> shows the fields qualified by the name of the table(s) they occur in. This means that key (connecting) fields will be listed more than once. (This alternative is only used for viewing purposes and has nothing to do with <i>Qualify</i> (page 264) fields in the load script.)</p> <p>It is also possible to view the fields of one table at a time. Note that available groups are always listed.</p>
<b>Edit Groups...</b>	<p>This button takes you directly to the <i>Document Properties: Groups</i> (page 442) page, where field groups to be used as dimensions can be defined.</p>
<b>Animate...</b>	<p>Opens <i>The Animation Dialog</i> (page 623), by which you can make use of the chart's first dimension for animation. Animation is only available for bitmap charts excluding pie charts. Some functional limitations apply when using animations.</p>
<b>Trellis...</b>	<p>Opens the <i>Trellis Settings</i> (page 625) dialog where you can create an array of charts based on the first dimension. Any type of bitmap chart can be made into a trellis display.</p>

**Used Dimensions**

This list contains the dimensions currently selected to be used as dimensions 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 expression.

**Background Color**

Double-click on **Background Color** in order to enter an attribute expression for calculating the cell background 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). This is done by using one of the special chart color functions (see *Color Functions (page 375)*). If the result of the expression is not a valid color representation, the program will default to black.

**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). This is done by using one of the special chart color functions (see *Color Functions (page 375)*). If the result of the expression is not a valid color representation, the program will default to black.

**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.

With the **Promote** and **Demote** buttons, dimensions in the **Used dimensions** list can be sorted.

**Add calculated dimension...**

Adds a new dimension and opens it for editing in the *Edit Expression Dialog (page 757)* dialog. A chart dimension is often in 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 *Advanced Aggregation (page 301)* function can be included for achieving nested aggregation.

**Edit...**

Opens the dimension for editing in the *Edit Expression Dialog (page 757)* dialog. See **Add calculated dimension...** above for details on calculated dimensions.

## Settings for Selected Dimension

In this group you find settings for individual dimensions.

### Enable Conditional

Marking this check box hides or shows the dimension dynamically, depending on the value of a condition expression entered, by clicking the ... button in the edit box below.

### Suppress When Value Is NULL

If this check box is enabled, the selected dimension in the **Used Dimensions** above will not be displayed in the chart if its value is NULL.

### Show All Values

Enable this check box to show all the dimension values regardless of the selection. As the expression value is zero for excluded dimension values, the option **Suppress Zero-Values** in the **Presentation** page must be deselected for **Show All Values** to work. **Show All Values** does not apply if you use an expression as dimension.

### Show Legend

When **Show Legend** is checked, the "names" of field values are shown along the x-axis.

### Label

With the **Label** option checked, the name of the field is shown. Labels can be edited in the text box below. A label can also be defined as a calculated label expression, see *Calculated Formula* (page 871) for dynamic update of the label text. Click on the ... button to open the *Edit Expression Dialog* (page 757) for easier editing of long formulas.

### Advanced...

This button opens the *Advanced Field Settings* (page 515) dialog which offers settings for image representation of field values and special text search options.

### Comment

A commentary field where the selected dimension can be described. The comment may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog* (page 757).

### Page Breaks

This setting only applies to the employment of page breaks in the [printout](#) from a pivot table or straight table. Three modes are available, with the following effects:

#### No Breaks

Will only insert page breaks at the end of each page, as required.

#### Conditional Breaks

Inserts a page break, unless all rows with the following dimension value can be fitted on the current page.

#### Forced Breaks

Inserts a page break whenever the dimension value changes

## The Animation Dialog

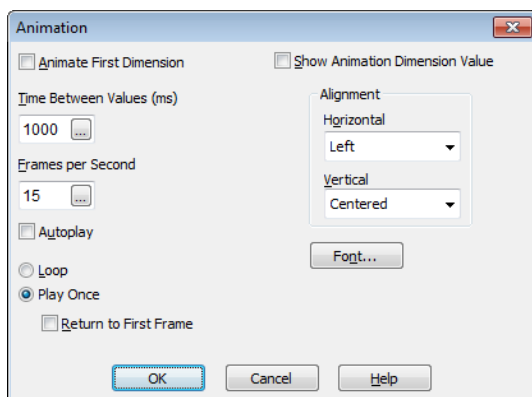
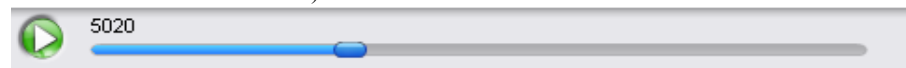


Chart animation is not supported when using the Ajax/WebView client.

**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 are more than one possible values 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. It is possible to stop or start the animation whenever you like using these controls. A **Progress Bar** shows the progress of the animation. It is possible to animate manually by pointing at the progress bar handle with the mouse, press down the left mouse button and drag 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 on the keyboard 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 Formula* (page 871).

**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 Formula* (page 871).

**Autoplay**

Enable this check box if an animation is to start automatically whenever a selection is made in the document.

**Loop**

Enable this option if you want the animation to play repeatedly until stopped with the **Pause** button in the **Animation Bar**.

**Play Once**

Enable this option if you want the animation to run only once from beginning to end, whenever it is started.

**Return to First Frame**

Enable this option if you want the animation to return to the first frame after completion.

**Show Animation Dimension Value**

By enabling this check box, the value of the data will be shown in the chart during the animation.

**Alignment**

Sets the alignment of the displayed value.

**Horizontal**

Aligns the value to the right, center or left.

**Vertical**

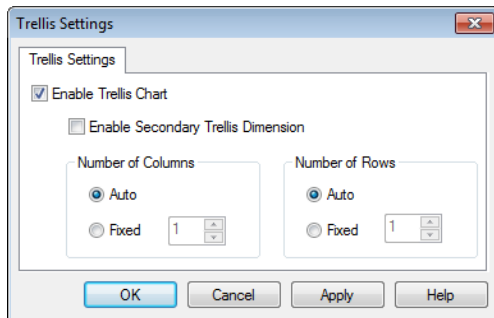
Aligns the value at the top, center or bottom.

**Font...**

Sets the font color for the displayed value.



## Trellis Settings



*Trellis Settings dialog*

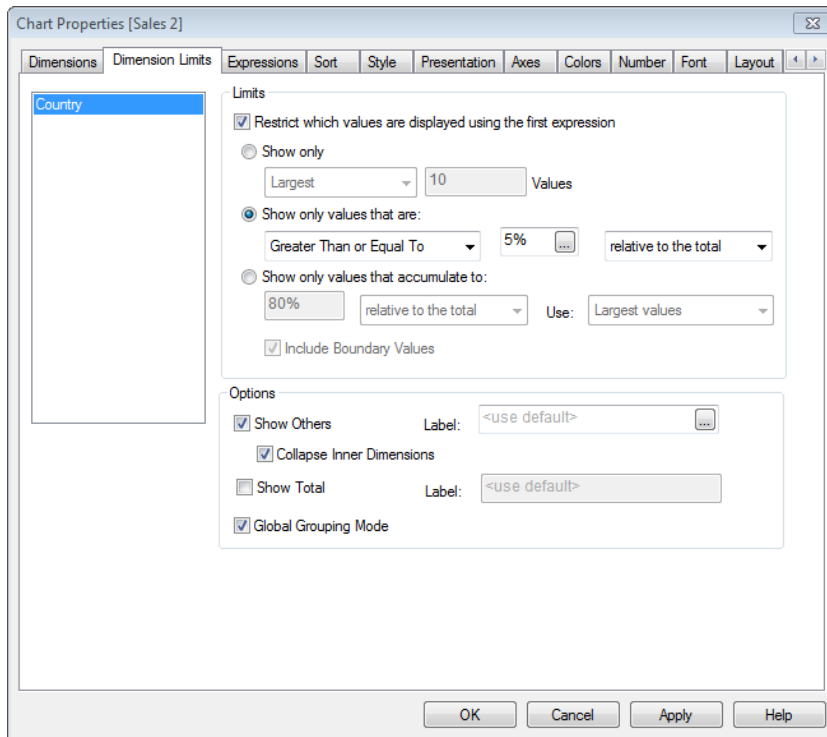
- Enable Trellis Chart** Enable this check box to create an array of charts based on the chart's first dimension.
- Enable Secondary Trellis Dimension** Enable this check box to include the second dimension in the trellis chart. If a secondary dimension is used, the first dimension values will be displayed 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

## 50.5 Chart Properties: Dimension Limits

Dimension Limits can be set for chart types, except for Gauge Charts and Pivot Tables.

The Dimension Limits tab controls the number of dimension values you can see in a given chart.

Before getting to that, it is important to describe the effect that the three options in the dropdown produce. The dropdown contains three values: **First**, **Largest** and **Smallest**. These values control the way the calculation engines sorts the values it returns to the charting engine. It is mandatory to have one of these options selected, if the dimension is to be restricted. The sorting only occurs for the first expression, except in pivot tables when the primary sort may override the first dimension sort.



*Chart Properties, Dimension Limits*

This property page is used for defining dimension limits. Each dimension in the chart is configured separately.

## Limits

### Restrict which values are displayed using the first expression

These properties are used to determine how many dimension values are displayed in the chart, according to settings made below.

#### Show only

Select this option if you want to show the **First**, **Largest** or **Smallest** x number of values. If this option is set to 5, there will be five values displayed. If the dimension has **Show Others** enabled, the Others segment will take up one of the five display slots.

The **First** option will return the rows based on the options selected on the **Sort** tab of the property dialog. If the chart is a Straight Table, the rows will be returned based on the primary sort at the time. In other words, a user can change the values display by double-clicking on any column header and making that column the primary sort.

The **Largest** option returns the rows in descending order based on the first expression in the chart. When used in a Straight Table, the dimension values shown will remain consistent while interactively sorting the expressions. The dimensions values will (may) change when the order of the expressions is changed.

The **Smallest** option returns the rows in ascending order based on the first expression in the chart. When used in a Straight Table, the dimension values shown will remain consistent while interactively sorting the expressions. The dimensions values will (may) change when the order of the expressions is changed.

Enter the number of values to display. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*.

#### Show only values that are:

Select this option to display all dimensions values that meet the specified condition for this option. Select to display values based on a percentage of the total, or on an exact amount. The **relative to the total** option enables a relative mode which is similar to the **Relative** option on the **Expressions** tab of the property dialog. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*.

#### Show only values that accumulate to:

When this option is selected, all rows up to the current row are accumulated, and the result is compared to the value set in the option. The **relative to the total** option enables a relative mode which is similar to the **Relative** option on the **Expressions** tab of the property dialog, and compares the accumulated values (based on first, largest or smallest values) to the overall total. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*. Select **Include Boundary Values** to include the dimension value that contains the comparison value.

#### Note!

Negative values will not be included when calculating the accumulated restriction sum. We recommend that you don't use relative restrictions for fields that can contain negative values.

#### Note!

If you add a limit to a calculated dimension and the data is sorted on the same calculated dimension, the dimension limit will be applied before the sort order is applied.

**Options****Show Others**

Enabling this option will produce an *Others* segment in the chart. All dimension values that do not meet the comparison criteria for the display restrictions will be grouped into the *Others* segment. If there are dimensions after the selected dimension, **Collapse Inner Dimensions** will control whether individual values for the subsequent / inner dimensions display on the chart.

**Label**

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 value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog* (page 757).

**Show Total**

The chart will display a total for the selected dimension when this option is enabled. This total behaves differently than the expression total, which is still configured on the **Expressions** tab of the property dialog.

**Label**

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 value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog* (page 757).

**Global Grouping Mode**

The option only applies to inner dimensions. When this option is enabled the restrictions will be calculated on the selected dimension only. All previous dimensions will be ignored. If this is disabled, the restrictions are calculated based on all preceding dimensions.

## Expression Totals Compared to Dimension Totals

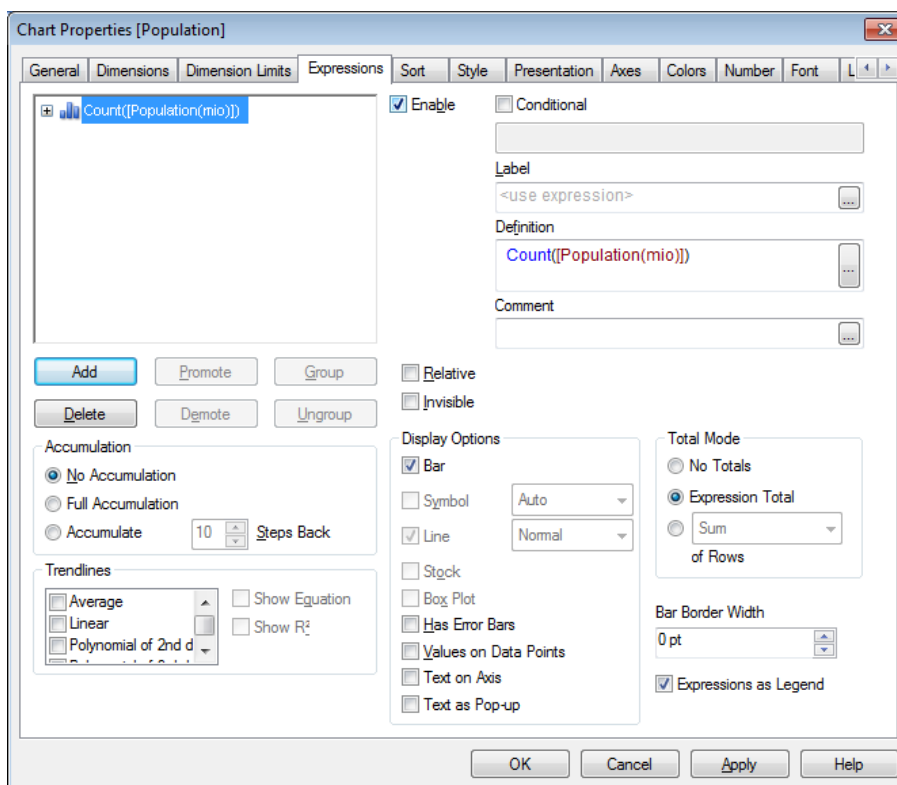
Dimension Totals are generated by the calculation engine, and are then returned to the charting engine as separate rows (or dimension values). This will have an impact on the Others rows. The difference between using Expression Totals and Dimension Totals can be seen below.

Expression Total			Dimension Total		
Region	Product Family	Sales	Region	Product Family	Sales
		<b>12,250,605</b>	<b>Overall Total</b>		<b>12,250,605</b>
Europe	Women's Clothes	3,138,666	<b>Europe</b>	<b>Total</b>	<b>7,753,390</b>
Europe	Men's Footwear	1,280,513	Europe	Women's Clothes	3,138,666
Europe	Sportswear	1,187,870	Europe	Men's Footwear	1,280,513
Europe	Men's Clothes	582,220	Europe	Sportswear	1,187,870
Europe	Women's Footwear	566,168	Europe	Men's Clothes	582,220
Europe	Baby Clothes	563,183	Europe	Women's Footwear	566,168
Europe	Children's Clothes	315,448	Europe	Baby Clothes	563,183
Europe	Swimwear	119,322	Europe	Children's Clothes	315,448
Europe	Swimwear	119,322	Europe	Swimwear	119,322
North America	Women's Clothes	707,181	<b>North America</b>	<b>Total</b>	<b>2,329,970</b>
North America	Sportswear	423,914	North America	Women's Clothes	707,181
North America	Women's Footwear	333,858	North America	Sportswear	423,914
North America	Men's Footwear	307,859	North America	Women's Footwear	333,858
North America	Men's Clothes	217,669	North America	Men's Footwear	307,859
North America	Children's Clothes	153,353	North America	Men's Clothes	217,669
North America	Baby Clothes	142,239	North America	Children's Clothes	153,353
North America	Swimwear	43,896	North America	Baby Clothes	142,239
Scandinavia	Women's Clothes	286,116	North America	Swimwear	43,896
Scandinavia	Sportswear	164,486			

### Expression Totals and Dimension Totals

When Dimension Totals is used, it is possible to have sub-totals within a straight table.

## 50.6 Chart Properties: Expressions



### Chart Properties, Expressions

To reach the **Chart Properties: Expressions** tab, right-click on a chart or table and select **Properties** from the **Object** menu.

When creating a chart, two questions should be asked:

- What should the size of the bars etc. illustrate? These are the **Expression(s)** ( e.g. **sum of NetSales**).
- How should the data be grouped? These are the **Dimension(s)** ( e.g. **per Country**).

---

**Note!**

The **Expressions** tab looks different for different chart and table types. If an option is grayed out, it is not available for that particular chart or table type.

---

## Expression List

The expression list in the top-left pane is a complete tree control with numerous control options.

In front of each expression (or expression group) an expansion icon (a box with a '+' ) is shown. Clicking the icon opens up underlying sub-expressions or attribute expressions. The icon is simultaneously replaced by a collapse icon ('-'). Certain plot options utilize sub-expressions, i.e. a set of two or more expressions that together define the plot symbol (e.g. **Stock** or **Box Plot** described below).

Expression data can also 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. These are:

### Background Color

Edit the default **Background Color** expression to create the attribute expression for calculating the plot color of the data point. The calculated color will have precedence over the default QlikView color selection and must be a valid color representation, which is achieved by using the *Color Functions* (page 375). If the result of the expression is not a valid color representation, the program will default to black. An auxiliary expression for **Text Color** can be created using the same method.

### Text Color

An auxiliary expression for **Text Color** can be created using the same method as for the background color (see above).

### Text Format

Edit the **Text Format** expression to enter an attribute expression for calculating the font style of text associated with the data point (For tables: text in the table cell for each dimension cell. The calculated text format will have precedence over table style defined in the *Chart Properties: Style* (page 755).) 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. Note that = is necessary before the string.

### Pie Popout

Click on the **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. This type of attribute expression only has effect on pie charts.

### Bar Offset

Click on **Bar Offset** in order to enter an attribute expression for calculating an offset for the bar or bar segment associated with the data point. The offset can be positive or negative and will move the bar or segment accordingly. This is useful e.g. when making so called waterfall charts. This type of attribute expression only has effect on bar charts.

### Line Style

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. This type of attribute expression only has effect on line, combo and radar charts.

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**

Click on **Show Value** in order to enter an attribute expression for calculating whether the data point plot should be complemented with a "value 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. This type of attribute expression only has effect on bar, line and combo charts.

**Add** New expressions and sub-expressions, are created by means of the **Add** button which takes you to the *Edit Expression Dialog (page 757)*. The option is also available in the context menu that appears when right-clicking in the list of expressions.

**Delete** The **Delete** button lets you remove previously created expressions from the list. The option is also available in the context menu that appears when right-clicking on an expression in the list of expressions.

**Copy** The **Copy** option is only available in the context menu that appears when right-clicking on an expression or a sub/attribute 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 **Export...** 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) 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 as** dialog from which you can choose the destination of the export file. The file will receive the extension Ex.xml.

**Paste** The **Paste** option is only available in the context menu that appears when right-clicking on an expression or sub/attribute expressions in the list of expressions. 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.



---

<b>Import</b>	The <b>Import</b> 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 expression. The imported expression will appear as a new expression in the chart.
<b>Promote Demote</b>	If several expressions are displayed, they can be sorted by means of the <b>Promote</b> and <b>Demote</b> buttons. This affects the order in which columns etc. are displayed in the chart.
<b>Group</b>	The <b>Group</b> button can be used for merging expressions into one or more cyclic groups, provided that two or more expressions are available. In the Qlik-View layout, you can cycle through the expressions belonging to one group by clicking the cycle icon that is displayed in the chart (= <b>Cycle Group</b> ). Right-click the same cycle icon to get a pop-up list of the expressions belonging to the group that are currently unused, for direct selection.
	<hr/> <b>Note!</b> Do not confuse <b>Cycle Group</b> with <i>Non-Hierarchical Groups (Cyclic)</i> (page 874)! <hr/>
<b>Ungroup</b>	Selecting an expression belonging to a group and clicking <b>Ungroup</b> , extracts the expression from the group. If only one expression remains in the cycle group after extraction, that last expression is also extracted and the group is removed.
<b>Enable</b>	Disabling this check box will set the expression to be omitted from the chart.
<b>Relative</b>	Enabling 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.
<b>Invisible</b>	Enabling this check box prevents the plotting of this expression while retaining the space allocated for it.
<b>Label</b>	In front of the expression label one or several icons are used for indicating the <i>Chart Types</i> (page 615) used and/or the <b>Display Options</b> selected for the expression (see below).
<b>Definition</b>	Shows the composition of the selected expression. It is possible to edit the expression directly in this box. By clicking the ... button the full <b>Edit Expression</b> dialog is opened.
<b>Comment</b>	This is a commentary field where the creator of the expression can describe the purpose and function of the expression.

**Display Options**

This group is used for modifying the way that data points are plotted or what will be entered in the expression cells of chart tables. Note that 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 its own icon in the Expressions list and appear as an empty placeholder with four sub expressions. The first sub expression will be used for plotting a high point of the stock marker. The second sub expression will be used for a low point. These two sub expressions must contain valid definitions in order for the stock marker to be drawn. The third sub expression is optional but is otherwise used for a close point of the stock marker. The fourth sub expression is also optional but is otherwise used for an open point of the stock marker. New empty sub expressions will be created automatically when **Stock** check box is marked for the expression. When **Stock** check box has been selected for an expression you cannot select **Bar**, **Line**, **Symbol**, **Box Plot** or **Has Error Bars** check boxes for the same expression. **Stock** check box cannot be selected for an expression if any of those options are already selected for the expression. This option is only available for combo charts.

**Box Plot**

Mark this check box to plot the expression as a box plot, often used for the display of statistical data. The expression will be preceded by its own icon in the Expressions list and appear as an empty placeholder with five sub expressions. The first sub expression will be used for plotting a box top point of the box plot. The second sub expression will be used for a box bottom point. These two expressions must contain valid definitions in order for the box plot to be drawn. The third to fifth sub expressions are optional. If used, those sub expressions define a median, an upper whisker, and a lower whisker. A common extension to a **Box Plot** is 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 the main expression. When **Box Plot** has been selected for an expression you cannot select **Bar**, **Line**, **Symbol**, **Stock** or **Has Error Bars** check boxes for the same

expression. **Box Plot** cannot be selected for an expression if any of those options are already selected for the expression. This option is only available for 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 their own 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 for bar, line and combo charts.

#### **Values on Data Point**

Mark this check box to have the result of an expression plotted as text on top of the data points. This option is only available for 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 an expression plotted as text at each x-axis value, the axis and the axis labels. This option is only available for bar, line and combo charts.

#### **Text as Pop-up**

Mark this check box to have the result of an 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.

**Display Options****Representation**

This option is only available for straight tables and pivot tables.

**Text**

The expression values are always interpreted and displayed as text.

**Image**

With this option QlikView will attempt 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 itself (e.g. qmem://<Name>/<Peter>). If QlikView cannot interpret an expression value as a valid image reference, the value itself will be displayed, unless the **Hide Text When Image Missing** box is checked.

**Circular Gauge, Linear Gauge, Traffic Light Gauge, LED Gauge**

With either gauge option, the gauge chart will be inscribed in the available table cell as an image. The layout of the gauge can be modified in the *Chart Properties: Presentation (Gauge Chart)* (page 722) dialog that is reached from the **Gauge Settings** button.

**Mini Chart**

With this option QlikView will display the expression values in a bar or line chart. The chart will be inscribed in the available table cell. The visual settings for the chart can be modified via the **Mini Chart Settings** button that opens the *Mini Chart Settings* (page 484) dialog. This option is only available for straight tables.

---

**Note!**

The mini chart will not be displayed when exporting to Excel!

---

**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.

**Image Formatting**

Only available when the **Image** option has been selected above. This option is only available for straight tables and pivot tables. 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 stretch-

ing. 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 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.

**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.

**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 the year 1997. If your chart contains several expressions, select the expression which values to be accumulated in the Expressions list. Accumulation is not available for pivot tables.

**No Accumulation**

If this option is selected, the y-values of the selected chart expression will not be accumulated.

**Full Accumulation**

If this option is selected, each y-value will accumulate 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**

This group is enabled for the selected expression for Straight Table chart objects only. There are three possible settings:

**No Totals**

Totals are not calculated for the selected expression.

**Expression Total**

The total of the expression evaluated on the next level. For example, if an expression generates the average monthly salary for a number of employees, the **Expression Total** will generate the total average of all the salaries.

**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 aggregated using the aggregation function selected from the drop-down list (typically summed up). **F(x) of Rows** is not available for pivot tables.

**Bar Border Width**

Specifies the width of the border line around bars plotted by this expression 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, this option displays a legend showing the expressions and their corresponding colors next to the chart.

**Trendlines**

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 **Trendlines** group 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 **Display 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.

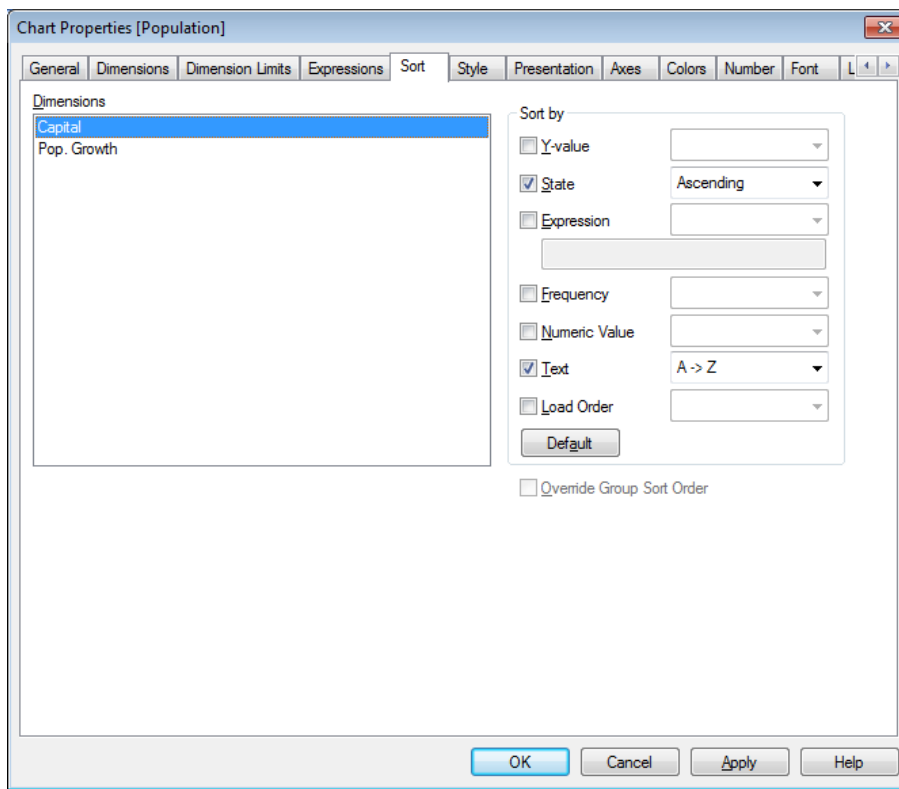
**Show Equation**

If this check box is marked for a specific expression, the expression's trend lines will be complemented by the trendline 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.

## 50.7 Chart Properties: Sort



*Chart Properties, Sort*

The **Chart Properties: Sort** page is reached by a right-click on a chart and selecting **Properties** from the **Object** menu.

This is where you decide the sort order of the chart dimension(s) from a number of available sort orders.

The **Dimensions** list contains the chart's dimensions. To assign a sort order, mark a dimension and choose one or more sort orders on the right side.

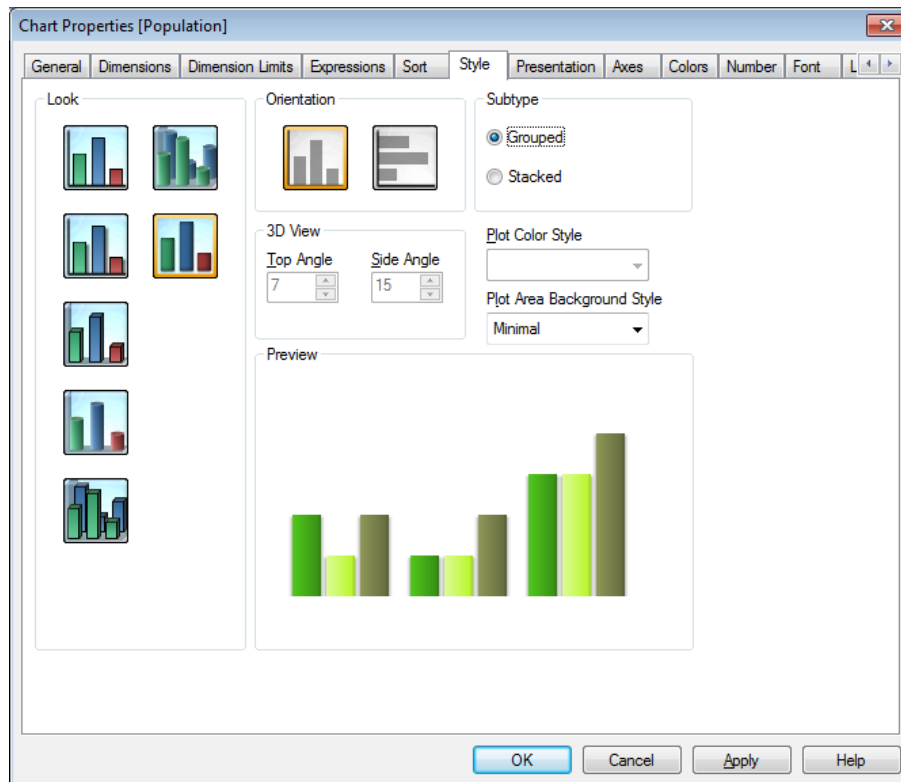
<b>Y-value</b>	Dimension values will be sorted by the numeric value of the y-axis. This option is not available for calculated dimensions.
<b>State</b>	Dimension values will be sorted according to their logical state, i.e. selected values before optional values, before excluded values.
<b>Expression</b>	Dimension values will be sorted according to the expression that is entered into the text edit box below this sort option.
<b>Frequency</b>	Dimension values will be sorted according to the number of occurrences in the table.
<b>Numeric Value</b>	Dimension values will be sorted according to their numeric value.
<b>Text</b>	Dimension values will be sorted according to their alphabetical order.
<b>Load Order</b>	Dimension values will be sorted according to their initial load order.

There is a hierarchy in the group from top to bottom so that when conflicting sort orders are selected, the first one encountered will take precedence. The selected sort order can be reversed by switching between **Ascending** and **Descending** or **A -> Z** and **Z -> A**.

By clicking the **Default** button, dimension values will be set to the default defined in the *Document Properties: Sort* (page 447) dialog.

The check box **Override Group Sort Order** 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 enabling this option 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.

## 50.8 Chart Properties: Style



*Chart Properties, Style*

On this page you can determine a basic style for the chart. Not all the listed features are available for every chart type.

- Look** Choose one of the available styles. This may in some instances affect not only the appearance of the chart but also its functionality.
- Orientation** Set the orientation of the chart, vertical or horizontal.



**Subtype**

In this group the mode is set, either **Grouped** or **Stacked** mode for bars (**Overlaid** or **Stacked** for Radar charts). This setting is only functional when the chart displays two dimensions or one dimension and 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.

Dimensions	Expressions	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 list all colors under **Color Map** on the **Colors** page will be changed to the selected style. The change is instantaneous and the setting itself will not be saved until the next time you enter this page of the dialog. The actual base colors in the color map are unaffected. The **Plot Color Style** is not available for all chart looks. The following 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 option gives a shadow effect on the plot area background.

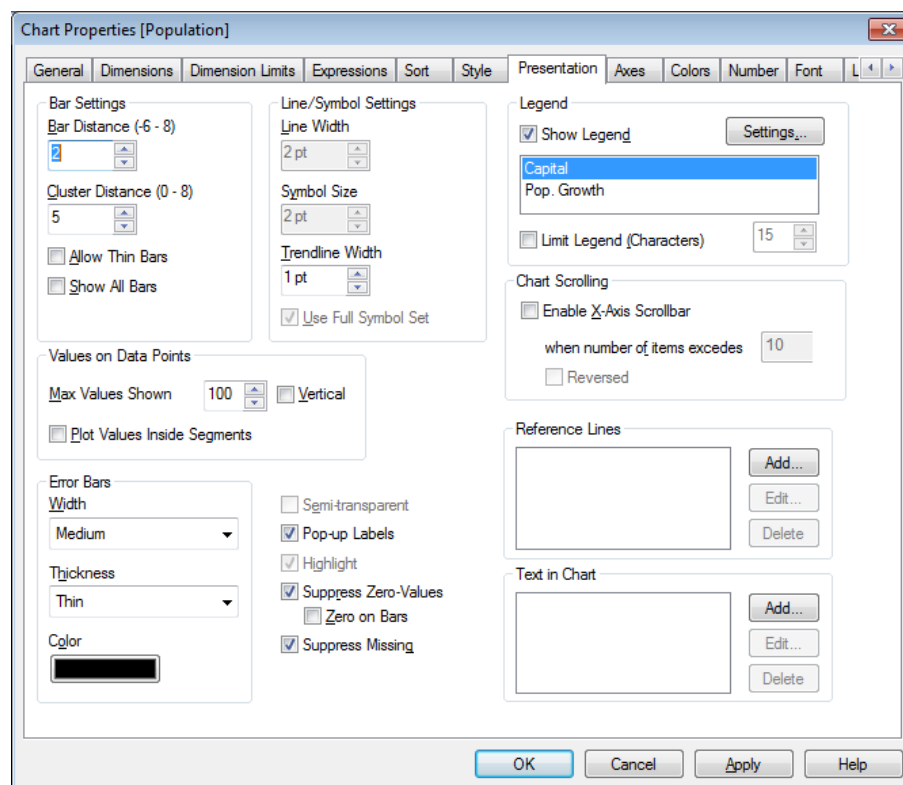
**Minimal**

This setting removes the plot area background.

**Preview**

Offers a preview of the basic visual properties of the chart.

## 50.9 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart)



*Chart Properties, Presentation*

This tab is used collectively for bar charts, line charts, combo charts, radar charts and mekko charts.

The **Bar Settings** group contains various display options for bars that are used in bar charts and combo charts.

- Bar Distance (-6 - 8)** Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.
- Cluster Distance (0 - 8)** Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.
- Allow Thin Bars** For charts with a 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. Bars are by default drawn with a minimum width of four pixels, to make them clearly distinguishable. Check this option to allow compression of bars to a width of 1 pixel.
- Show All Bars** For charts with a 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. Check this option to force the drawing of all data points. Bars can be compressed (as for **Allow Thin Bars**) and some may also be partially obscured by others.

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 **Display Options** in *Chart Properties: Expressions (page 629)* page.

**Max Values Shown** In this box you can specify an upper limit for the number of 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.

**Vertical** Shows the values vertically.

**Plot Values Inside Segments** Marking this check box will plot values on data points inside the segments instead of on top of them.

In the **Error Bars** group display options for any error bars used in the chart is determined

**Width** Specifies the width of error bars.

**Thickness** Specifies the thickness of error bars.

**Color** Sets a color for error bars.

In the **Line/Symbol Settings** group display options for lines and data point symbols that are used in line charts and combo charts are determined. It is also possible to determine the width of trendlines.

**Line Width** Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

**Symbol Size** Determines the size of symbols, if a symbol representation is specified.

**Trendline Width** This setting determines the width of trendlines.

**Use Full Symbol Set** This alternative makes more symbol representation available (rings, triangles etc.)

**Semi-transparent** Check this option if you want filled lines to be drawn semi-transparent.

**Pop-up Labels** Check this option to display the corresponding dimension value in a pop-up window when the mouse pointer touches a value.

**Highlight** With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.

**Suppress Zero-Values** This check box eliminates dimensions that are empty or contain only zeros. This option is selected by default.

**Zero on Bars**

This option is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Points** is selected for the chart expression under **Display Options** in *Chart Properties: Expressions (page 629)* , 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.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

- Show Legend** Check this alternative to include a legend in the chart (checked by default). It is possible to change the *Legend Settings (page 646)* by clicking the **Settings...** button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.
- Limit Legend (Characters)** Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

- Enable X-Axis Scrollbar** Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under **When Number of Items Exceeds**.
- Reversed** Checking the box displays the values in reversed order.

In the **Reference Lines** group you can define reference (grid) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

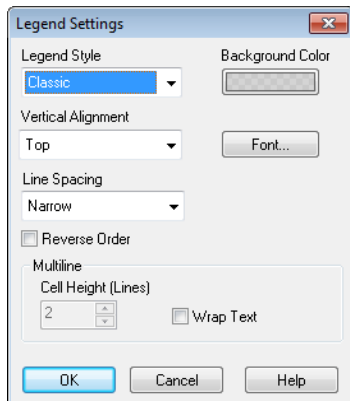
- Add** Opens the *Reference Lines (page 647)* dialog where you can create a new reference line in the chart.
- Edit** Highlight an existing reference line in the list and click this button to edit its properties in the *Reference Lines (page 647)* dialog.
- Delete** Highlight an existing reference line in the list and click this button to delete it from the list.

The **Text in Chart** group is used for adding free-floating text to the chart.

- Add** Opens the *Chart Text (page 648)* dialog, where you can create a new chart text.
- Edit** Highlight an existing text in the list and click this button to edit its properties in the *Chart Text (page 648)* dialog.
- Delete** Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in the layout edit mode. See *Sizing and Moving Chart Components (page 612)*.

## Legend Settings

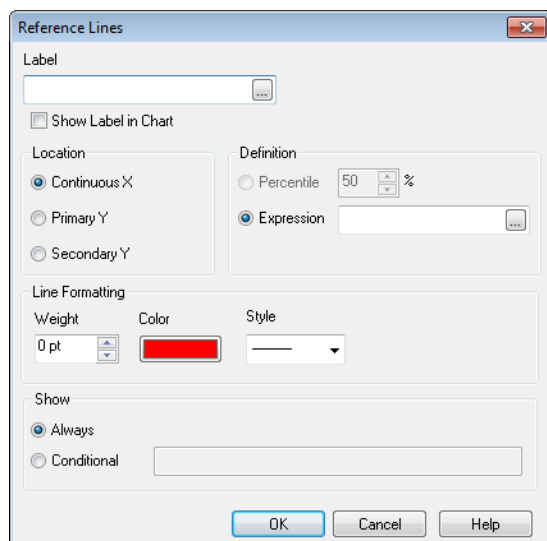


*The Legend Settings dialog*

The layout of the chart legend is controlled by the various settings of this dialog.

<b>Legend Style</b>	Sets the basic style for the legend. Choose between several styles.
<b>Background Color</b>	Sets the color of the legend background. The color can be defined as a solid color or as a gradient via the <i>Color Area (page 428)</i> dialog that opens when clicking the button.
<b>Vertical Alignment</b>	Specifies how the legend is positioned in relation to the plot area, when it needs less vertical space than the plot area.
<b>Font</b>	Opens the standard <i>Font (page 491)</i> dialog where a font for the legend can be specified.
<b>Line Spacing</b>	Specifies the distance between items in the legend.
<b>Reverse Order</b>	Reverses the sort order of the legend.
<b>Multiline</b>	Sets the options for multi line legend items: <ul style="list-style-type: none"> <li><b>Wrap Text</b> Wraps the text of the legend items in two or more lines.</li> <li><b>Cell Height (Lines)</b> If the <b>Wrap Text</b> option is enabled, this setting specifies how many lines that should be used for each item.</li> </ul>

## Reference Lines



The Reference Lines dialog for Bar chart.

The dialog may vary slightly in appearance depending on the kind of chart that is used. A reference line is a line intersecting the chart plot area from a given point on one or both axes. It may be used, for example, 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 originates.

- Label** Enter a label to be drawn next to the reference line. The default value used is the expression. The label may be defined as a calculated expression.
- Show Label in Chart** Enable this setting if the label should appear next to the reference line.
- Location** Sets from which axis the reference line should originate:
- Continuous X**  
The reference line originates from the x-axis. This option is only available if the chart has a continuous x-axis (see *Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart)* (page 649)).
  - Primary Y**  
The reference line originates from the primary y-axis (left/bottom).
  - Secondary Y**  
The reference line originates from the secondary y-axis (right/top).
- Definition** Sets the value at which the reference line should be drawn. The value be either a fixed **Percentile** (enter a value between 1 and 100 in the edit box) of the current chart data or an arbitrary numeric **Expression**.

**Line Formatting**

Defines the layout of the reference line:

**Weight**

Specifies the weight of the reference line. The value can be specified in mm, cm, inches ("), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

**Color**

Sets the color for the reference line.

**Style**

Specifies the style for the reference line, for example, continuous, dashed or dotted.

**Show**

Specifies the condition for displaying the reference line.

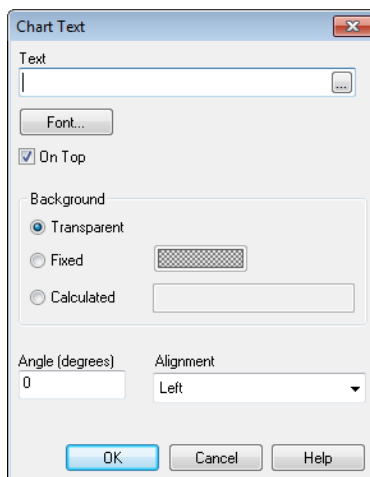
**Always**

The reference line will always be displayed.

**Conditional**

The reference line will be displayed 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 Text



*The Chart Text dialog*

**Text**

Enter a text that should be displayed in the chart. The text entered can also be defined as *Calculated Formula* (page 871) for dynamic update. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas or for typing multi line text.

**Font**

Opens the standard *Font* (page 491) dialog where a font for the text can be specified.

**On Top**

Forces the text to the foreground when the chart is drawn.



**Background**

Defines the background of the text.

**Transparent**

With this option, only the text itself will be visible. Any sheet object covered by the text will be fully visible.

**Fixed**

This option lets you pick a background color by clicking the **Color** button to the right of the radio button.

**Calculated**

The background color may be dynamically calculated from an expression. The expression must be a valid color representation, which is achieved by using the *Color Functions* (page 375). Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas. If the result of the expression is not a valid color representation, the program will default to black.

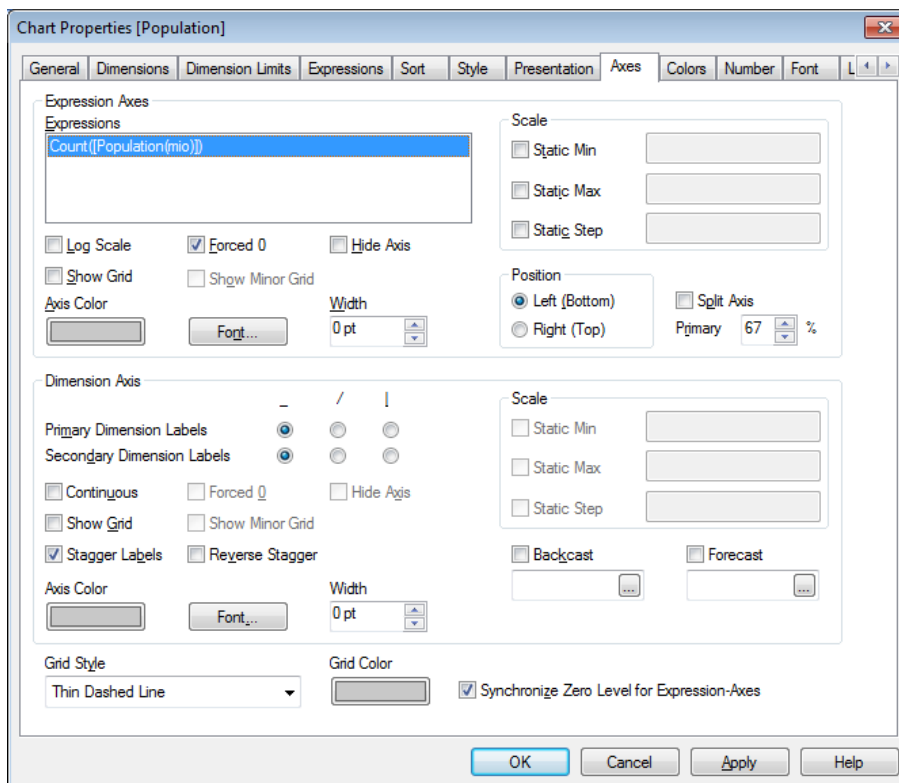
**Angle (degrees)**

Specifies the angle for the text. 0 to 360 degrees is allowed, the default value is 0.

**Alignment**

Sets the horizontal alignment of the text within its background.

## 50.10 Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart)



*Chart Properties, Axes*

This dialog is opened from the **Chart Properties** dialog of bar charts, line charts, combo charts and radar charts.

In this dialog the appearance of the axes and the expressions, represented by the y-axes of the chart, can be set. One or two y-axis scales may be defined. In case the x-axis represents numeric values it may be set to **continuous** (see below).

The contents of the dialog:

## Expression Axes

**Expressions** The available expressions displayed here are defined in the *Chart Properties: Expressions (page 629)* dialog.

Set a separate **Axis Color** and **Width** for the y-axis as well as a separate **Font** for the scale numbering, by clicking the appropriate options.

**Log Scale** A logarithmic scale can be used, provided that all data points in the chart have positive values ( $>0$ ).

**Forced 0** The x-axis will cross at  $y = 0$ . This option is not available when a logarithmic axis is used.

**Hide Axis** Hides the axis for the selected expression.

**Show Grid / Show Minor Grid** The scale ticks of the y-axis that will be used for showing horizontal (y-axis **Position** is set to **Left** and/or **Right**) and vertical (y-axis **Position** is set to **Top** and/or **Bottom**) grid lines.

**Scale**

**Static Min**  
The scale of the y-axis will not change with document state. Check this alternative to set a fixed minimum value for the y-axis in the edit box.

**Static Max**  
Check this alternative to set a fixed maximum value for the y-axis.

**Static Step**  
Check this alternative to set a fixed interval between scale ticks for the y-axis.

Values entered in the **Scale** group may be specified as a *Calculated Formula (page 871)*. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

**Position** When two expressions are available, you may select different positions for them so that one is displayed to the **Left (Bottom)** and the other is displayed to the **Right (Top)**. In this way the y-axes will display different scales for the expressions.

**Split Axis** Splitting the y-axis into two parts creates the impression of two charts sharing a common x-axis. The **Primary  $n$  %** setting defines the percentage of the available axis length that will be used for the primary part of the axis.

## Dimension Axis

<b>Primary Dimension Labels</b>	Set the display of labels for main dimension as either horizontal, diagonal or vertical text.
<b>Secondary Dimension Labels</b>	Display labels for secondary dimension as either horizontal, diagonal or vertical text.
<b>Continuous</b>	Scale axes as continuous numeric (linear).
<b>Forced 0</b>	The y-axis will cross at $x = 0$ .
<b>Hide Axis</b>	The x-axis will not be shown.
<b>Show Grid</b>	The tick marks of the x-axis will be extended into grid lines. The Grid layout can be modified via the <b>Grid Style</b> and <b>Grid Color</b> settings (See end of page).
<b>Show Minor Grid</b>	This alternative enables a subdivision of the grid.
<b>Stagger Labels</b>	When there is not enough room for displaying the labels of all x-axis values, labels will be staggered. When this option is deselected, fewer labels may be displayed. This setting only affects horizontal labels. Labels are normally staggered from left to right. If you mark the check box <b>Reverse Stagger</b> , the stagger will be reversed to right to left.

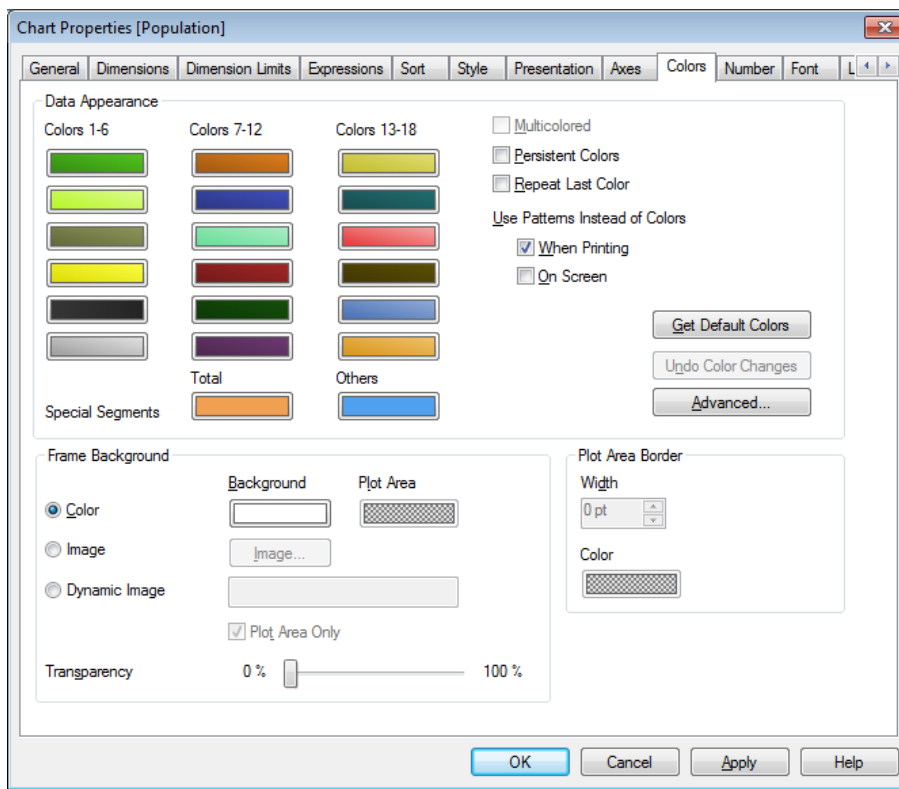
Set a separate **Axis Color** and **Width** for the x-axis as well as a separate **Font** for the scale numbering, by clicking the appropriate alternatives.

<b>Scale</b>	<p><b>Static Min</b> The scale of the x-axis will not change with document state. Check this alternative to set a fixed minimum value for the x-axis in the edit box.</p> <p><b>Static Max</b> Check this alternative to set a fixed maximum value for the x-axis.</p> <p><b>Static Step</b> Check this alternative to set a fixed interval between scale ticks for the x-axis.</p>
--------------	---

Values entered in the **Scale** group may be specified as a *Calculated Formula* (page 871). Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

<b>Backcast</b>	This function operates on trend lines. Enter into the text edit box how far back you would like to estimate the trend line. See <i>Chart Properties: Expressions</i> (page 629). The backcast portion of trend lines is displayed as dotted.
<b>Forecast</b>	Enter into the text edit box how far you wish to predict the trend line. See <i>Chart Properties: Expressions</i> (page 629). The forecast portion of trend lines is displayed as dotted.
<b>Grid Style</b>	With <b>Show Grid</b> checked, you can choose from the available grid styles in the drop-down list.
<b>Grid Color</b>	This alternative lets you pick a color for the grid.
<b>Synchronize Zero Level for Expression-Axes</b>	When two y-axes are displayed, this setting is used for synchronizing the zero level of them.

## 50.11 Chart Properties: Colors



*Chart Properties, Colors*

The **Chart Properties: Colors** page is opened by right-clicking a chart window and choosing the **Properties** command from the **Object** menu.

In the **Data Appearance** group it is possible to assign up to 18 different colors to the dimension field values of charts.

### **Colors 1 - 18**

Colors can be defined as solid colors or color gradients. To customize a color, click its button to open the *Color Area* (page 428) dialog.

The **Get Default Colors** button resets the color map to the default settings of QlikView.

The **Undo Color Changes** button returns the color settings that applied upon entry to this dialog.

The **Advanced...** button opens the *Advanced Color Map* (page 653) dialog where color maps can be set and retrieved on sheet, document, user and Qlik-View default level.

### **Multicolored**

Unchecking this option results in all bars having the same color.

### **Persistent Colors**

Checking this option locks the color map so that each value has a color permanently assigned to it.

### **Repeat Last Color**

Checking this option assigns the 18th color in the color map to any value after the 18th value. If unchecked, colors are just repeated in succession from 1 to 18.

### Use Patterns Instead of Colors

#### When Printing

The chart will be printed in black and white, with bar areas hatched. If unchecked, a monochrome printer will use gray scaling instead.

#### On Screen

Displays the chart with hatched bar areas.

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.

Clicking either button opens the *Color Area (page 428)* dialog.

Tip! The **Background Color** setting can be combined with the **Image** and/or **Plot Area Only** options below.

#### 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

Select this alternative and click the **Image** button to open the **Select Image** dialog, where it is possible to import a background picture.

Limit the imported picture to the **Plot Area Only** by checking this alternative.

#### Dynamic Image

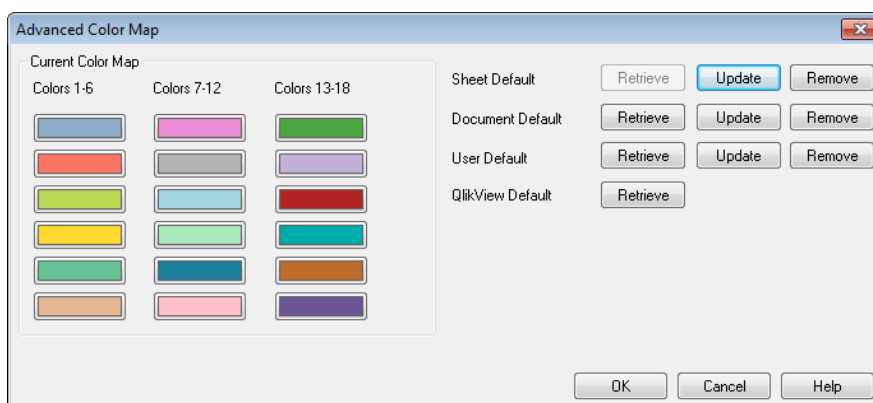
Enter a calculated expression to show dynamic background images that change with the selection. Available for bar, line, combo, scatter and grid charts.

#### Transparency

Sets the degree of transparency of the chart background. At 0% the background will be completely opaque with the color defined under **Background Color** above. At 100% the background will be completely transparent.

In the **Plot Area Border** group you can assign **Width** and **Color** for the rectangle surrounding the plot area.

## Advanced Color Map

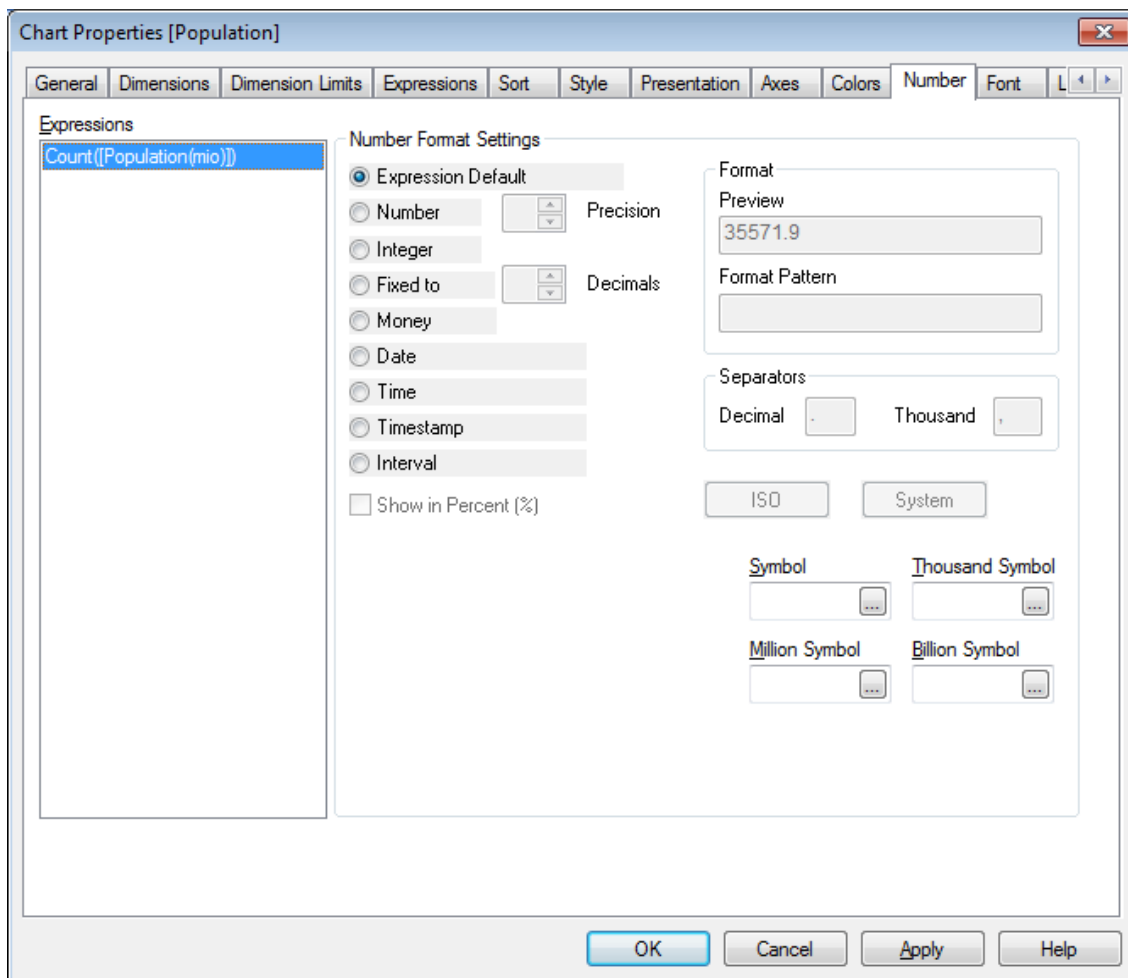


*The Advanced Color Map dialog*

In the this dialog, color settings for several different levels can be retrieved, set or cleared.

<b>Current Color Map</b>	When the dialog is opened, the color map of the current chart is displayed. Change the color map by clicking individual colors or by pressing the <b>Retrieve</b> button for <b>Sheet Default</b> , <b>Document Default</b> , <b>User Default</b> or <b>Qlik-View Default</b> .
<b>Sheet Default</b>	Sets, retrieves or clears the default color map for the current sheet. Press <b>Retrieve</b> to retrieve the current sheet default color map (the option is only available if a sheet default is available). Press <b>Update</b> to apply the current color map as sheet default. Press <b>Remove</b> to clear the default for the current sheet (the option is only available if a sheet default is available).
<b>Document Default</b>	Sets, retrieves or clears the default color map for the current document. Press <b>Retrieve</b> to retrieve the current document default color map (the option is only available if a document default is available). Press <b>Update</b> to apply the current color map as document default. Press <b>Remove</b> to clear the default for the current document (the option is only available if a document default is available).
<b>User Default</b>	Sets, retrieves or clears the default color map for the current user. Press <b>Retrieve</b> to retrieve the current user default color map (the option is only available if a user default is available). Press <b>Update</b> to apply the current color map as user default. Press <b>Remove</b> to clear the default for the current user (the option is only available if a user default is available).
<b>QlikView Default</b>	Press <b>Retrieve</b> to retrieve the QlikView default color map. This default cannot be changed.

## 50.12 Chart Properties: Number



*Chart Properties, Number*

This property page applies to the active chart and contains the following controls for formatting values:

- |                           |   |
|---------------------------|---|
| <b>Expression Default</b> | Shows numeric values using the number format provided by the expression.  |
| <b>Number</b>             | Shows numeric values with the number of digits set in the <b>Precision</b> spinner box.   |
| <b>Integer</b>            | Shows numeric values as integers.   |
| <b>Fixed to</b>           | Shows numeric values as decimal values with the number of decimal digits set in the <b>Decimals</b> spinner box.  |
| <b>Money</b>              | Shows numeric values in the format shown in the <b>Preview</b> text box. The default format is the Windows Currency setting.  |
| <b>Date</b>               | Shows values that can be interpreted as dates in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box. |
| <b>Time</b>               | Shows values that can be interpreted as time in the format set in the <b>Format Pattern</b> edit box. An example of this format is shown in the <b>Preview</b> text box.  |

**Timestamp** Shows values that can be interpreted as date + time in the format set in the **Format Pattern** edit box. An example of this format is shown in the **Preview** text box.

**Interval** Shows time as sequential time increment (e.g. format = *mm* shows the value as the number of minutes since calendar start ( 1899:12:30:24:00)).

The **Show in Percent (%)** button operates on the following formats: **Number**, **Integer** and **Fixed to**.

**Decimal** and **Thousand** separators can be set in the edit boxes of the **Separators** group.

In the **Symbol** edit boxes symbols for unit, 1000, 1000 000 and 1000 000 000 can be entered.

The **ISO** button sets the time, date and timestamp formatting to ISO standard.

The **System** button sets the formatting to system settings.

## 50.13 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 50.14 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

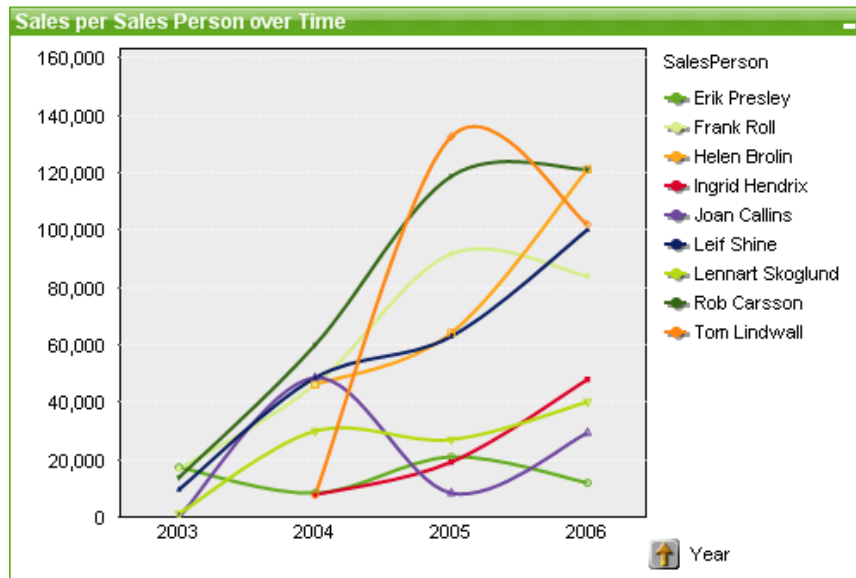
## 50.15 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



# 51 Line Chart



An example of a Line chart

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

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.

The quickest way to create a new line chart is to select *Quick Chart Wizard* (page 767) from the **Tools** menu.

By a right-click on the line chart the *Line Chart: Object Menu* (page 658) will be displayed. It can also be accessed from the **Object** menu, when the line chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

## 51.1 New Chart

A new chart can be created by clicking on the **Create Chart** tool in the toolbar, by a right-click on an empty sheet area and selecting **New Sheet Object, Chart** from **Object** menu or by selecting **New Sheet Object, Chart** from **Layout** menu. This will open a sequence of chart property pages.

In the page that appears first, name and type of graph and title (optional) is set, clicking the **Next** button opens the second page etc. As soon as sufficient information has been entered the **Next** and/or **Finish** buttons are enabled and the user can proceed to the next page in the sequence or finish it.

Once the chart is displayed on the sheet it can be modified by right clicking the graph and selecting **Properties** or by activating the graph (click on caption area) and selecting **Properties** in the **Object** menu.


## 51.2 Line Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.



The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments (page 472)</i> for further information.
<b>Detach</b>	<p>The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached.</p> <p>By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.</p>
<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
<b>Set Reference</b>	<p>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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.</p>
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.

---

<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.
<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Values</b> Copies the values to the clipboard in the form of a table.</p> <p><b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.

---

<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 51.3 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 51.4 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 51.5 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 51.6 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

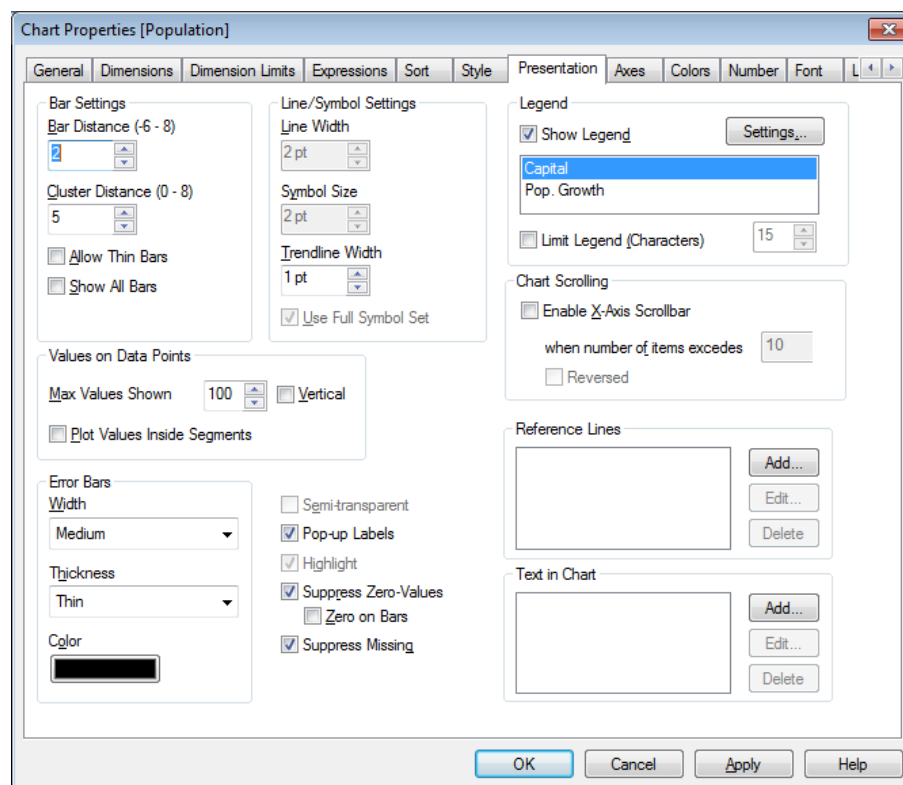
## 51.7 Sort

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. See *Chart Properties: Sort (page 639)*.

## 51.8 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 51.9 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart)



*Chart Properties, Presentation*

This tab is used collectively for bar charts, line charts, combo charts, radar charts and mekko charts.

The **Bar Settings** group contains various display options for bars that are used in bar charts and combo charts.

- Bar Distance (-6 - 8)** Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.
- Cluster Distance (0 - 8)** Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.
- Allow Thin Bars** For charts with a 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. Bars are by default drawn with a minimum width of four pixels, to make them clearly distinguishable. Check this option to allow compression of bars to a width of 1 pixel.
- Show All Bars** For charts with a 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. Check this option to force the drawing of all data points. Bars can be compressed (as for **Allow Thin Bars**) and some may also be partially obscured by others.

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 **Display Options** in *Chart Properties: Expressions (page 629)* page.

**Max Values Shown** In this box you can specify an upper limit for the number of 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.

**Vertical** Shows the values vertically.

**Plot Values Inside Segments** Marking this check box will plot values on data points inside the segments instead of on top of them.

In the **Error Bars** group display options for any error bars used in the chart is determined

**Width** Specifies the width of error bars.

**Thickness** Specifies the thickness of error bars.

**Color** Sets a color for error bars.

In the **Line/Symbol Settings** group display options for lines and data point symbols that are used in line charts and combo charts are determined. It is also possible to determine the width of trendlines.

**Line Width** Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

**Symbol Size** Determines the size of symbols, if a symbol representation is specified.

**Trendline Width** This setting determines the width of trendlines.

**Use Full Symbol Set** This alternative makes more symbol representation available (rings, triangles etc.)

**Semi-transparent** Check this option if you want filled lines to be drawn semi-transparent.

**Pop-up Labels** Check this option to display the corresponding dimension value in a pop-up window when the mouse pointer touches a value.

**Highlight** With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.

**Suppress Zero-Values** This check box eliminates dimensions that are empty or contain only zeros. This option is selected by default.

**Zero on Bars**

This option is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Points** is selected for the chart expression under **Display Options** in *Chart Properties: Expressions (page 629)* , 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.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

- |                                  |   |
|----------------------------------|---|
| <b>Show Legend</b>               | Check this alternative to include a legend in the chart (checked by default). It is possible to change the <i>Legend Settings (page 646)</i> by clicking the <b>Settings...</b> button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead. |
| <b>Limit Legend (Characters)</b> | Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.   |

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

- |                                |  |
|--------------------------------|--|
| <b>Enable X-Axis Scrollbar</b> | Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under <b>When Number of Items Exceeds</b> . |
| <b>Reversed</b>                | Checking the box displays the values in reversed order.  |

In the **Reference Lines** group you can define reference (grid) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

- |               |  |
|---------------|--|
| <b>Add</b>    | Opens the <i>Reference Lines (page 647)</i> dialog where you can create a new reference line in the chart.                                     |
| <b>Edit</b>   | Highlight an existing reference line in the list and click this button to edit its properties in the <i>Reference Lines (page 647)</i> dialog. |
| <b>Delete</b> | Highlight an existing reference line in the list and click this button to delete it from the list.   |

The **Text in Chart** group is used for adding free-floating text to the chart.

- |               |   |
|---------------|---|
| <b>Add</b>    | Opens the <i>Chart Text (page 648)</i> dialog, where you can create a new chart text.   |
| <b>Edit</b>   | Highlight an existing text in the list and click this button to edit its properties in the <i>Chart Text (page 648)</i> dialog. |
| <b>Delete</b> | Highlight an existing text in the list and click this button to delete it from the list.  |

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in the layout edit mode. See *Sizing and Moving Chart Components (page 612)*.

## 51.10 Axes

On the **Axes** page you can set the display properties for the x- and y-axes.

This page is identical to *Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart) (page 649)*.

## 51.11 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 51.12 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 51.13 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 51.14 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

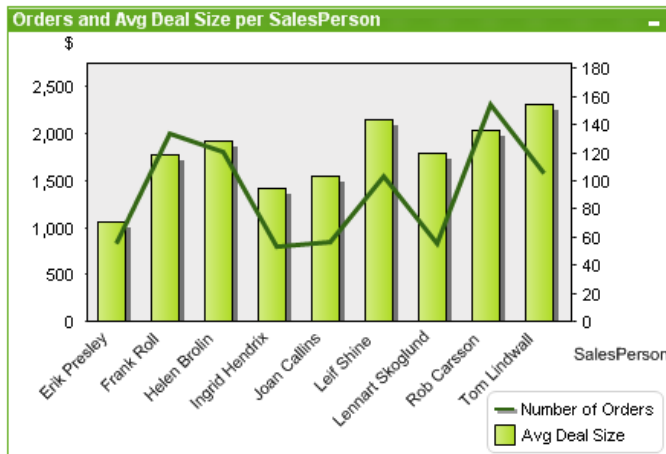
## 51.15 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.




## 52 Combo Chart



An example of a Combo chart

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

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 another expression as a line or symbols.

The easiest way to create a combo chart is to click the **Create Chart**  button in the toolbar.

By a right-click on the combo chart the *Combo Chart: Object Menu* (page 665) will be displayed. It can also be accessed from the **Object** menu, when the combo chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).




### 52.1 Combo Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you can make direct comparisons between the copy and the original.

<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General</i> (page 91) dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.

<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Values</b> Copies the values to the clipboard in the form of a table. <b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 52.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 52.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 52.4 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 52.5 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

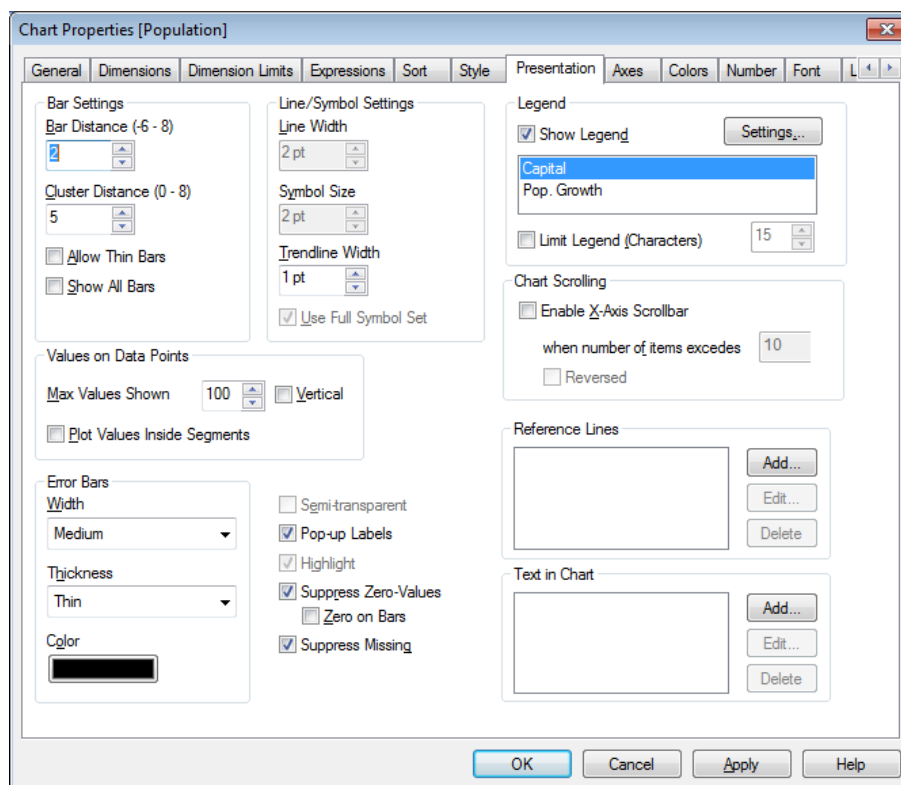
## 52.6 Sort

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. See *Chart Properties: Sort (page 639)*.

## 52.7 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 52.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart)



*Chart Properties, Presentation*

This tab is used collectively for bar charts, line charts, combo charts, radar charts and mekko charts.

The **Bar Settings** group contains various display options for bars that are used in bar charts and combo charts.

**Bar Distance (-6 - 8)** Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.

- Cluster Distance (0 - 8)** Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.
- Allow Thin Bars** For charts with a 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. Bars are by default drawn with a minimum width of four pixels, to make them clearly distinguishable. Check this option to allow compression of bars to a width of 1 pixel.
- Show All Bars** For charts with a 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. Check this option to force the drawing of all data points. Bars can be compressed (as for **Allow Thin Bars**) and some may also be partially obscured by others.

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 **Display Options** in *Chart Properties: Expressions (page 629)* page.

- Max Values Shown** In this box you can specify an upper limit for the number of 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.
- Vertical** Shows the values vertically.
- Plot Values Inside Segments** Marking this check box will plot values on data points inside the segments instead of on top of them.

In the **Error Bars** group display options for any error bars used in the chart is determined

- Width** Specifies the width of error bars.
- Thickness** Specifies the thickness of error bars.
- Color** Sets a color for error bars.

In the **Line/Symbol Settings** group display options for lines and data point symbols that are used in line charts and combo charts are determined. It is also possible to determine the width of trendlines.

- Line Width** Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).
- Symbol Size** Determines the size of symbols, if a symbol representation is specified.
- Trendline Width** This setting determines the width of trendlines.
- Use Full Symbol Set** This alternative makes more symbol representation available (rings, triangles etc.)
- Semi-transparent** Check this option if you want filled lines to be drawn semi-transparent.
- Pop-up Labels** Check this option to display the corresponding dimension value in a pop-up window when the mouse pointer touches a value.

**Highlight** With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.

**Suppress Zero-Values** This check box eliminates dimensions that are empty or contain only zeros. This option is selected by default.

**Zero on Bars**

This option is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Points** is selected for the chart expression under **Display Options** in *Chart Properties: Expressions (page 629)*, 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.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

**Show Legend** Check this alternative to include a legend in the chart (checked by default). It is possible to change the *Legend Settings (page 646)* by clicking the **Settings...** button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.

**Limit Legend (Characters)** Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

**Enable X-Axis Scrollbar** Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under **When Number of Items Exceeds**.

**Reversed** Checking the box displays the values in reversed order.

In the **Reference Lines** group you can define reference (grid) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

**Add** Opens the *Reference Lines (page 647)* dialog where you can create a new reference line in the chart.

**Edit** Highlight an existing reference line in the list and click this button to edit its properties in the *Reference Lines (page 647)* dialog.

**Delete** Highlight an existing reference line in the list and click this button to delete it from the list.

The **Text in Chart** group is used for adding free-floating text to the chart.

**Add** Opens the *Chart Text (page 648)* dialog, where you can create a new chart text.

<b>Edit</b>	Highlight an existing text in the list and click this button to edit its properties in the <i>Chart Text (page 648)</i> dialog.
<b>Delete</b>	Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in the layout edit mode. See *Sizing and Moving Chart Components (page 612)*.

## 52.9 Axes

On the **Axes** page you can set the display properties for the x- and y-axes.

This page is identical to *Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart) (page 649)*.

## 52.10 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 52.11 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 52.12 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 52.13 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 52.14 Caption

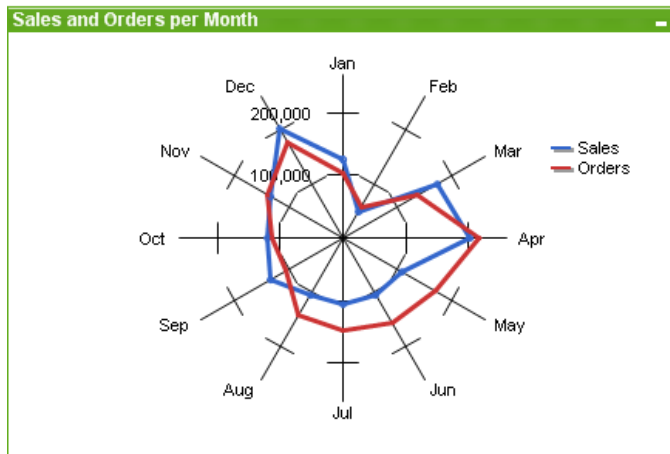
On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.





## 53 Radar Chart



*An example of a Radar chart*

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

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.

The easiest way to create a radar chart is to click the **Create Chart**  button in the toolbar.

By a right-click on the radar chart the *Radar Chart: Object Menu* (page 673) will be displayed. It can also be accessed from the **Object** menu, when the radar chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).




### 53.1 Radar Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.

<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General</i> (page 91) dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.

<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Values</b> Copies the values to the clipboard in the form of a table. <b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 53.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 53.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 53.4 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 53.5 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

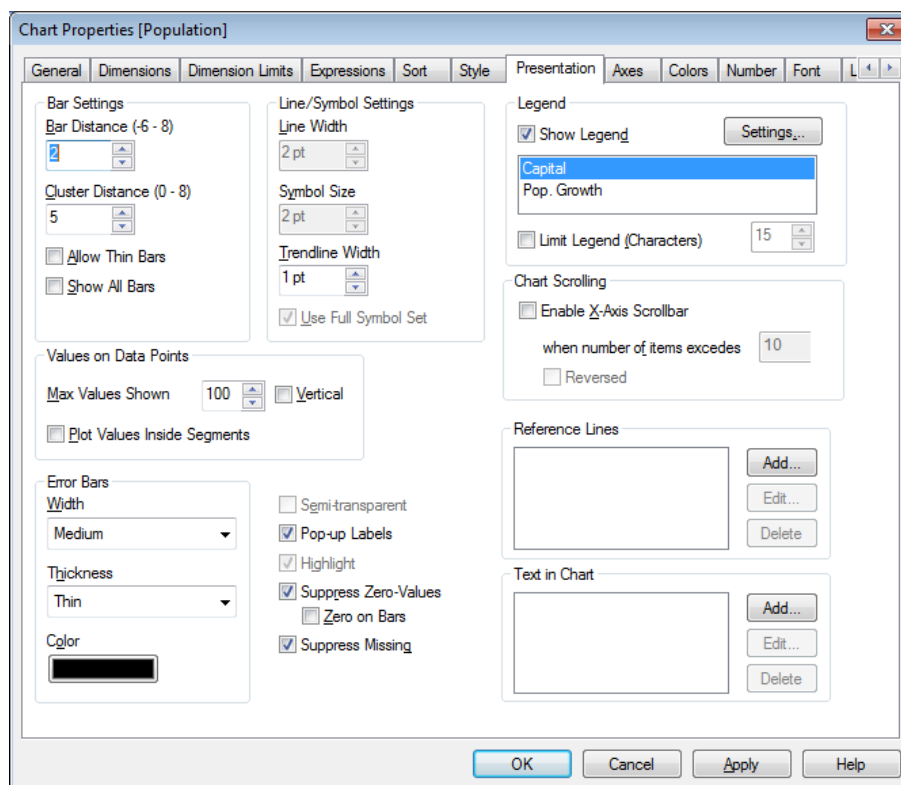
## 53.6 Sort

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. See *Chart Properties: Sort (page 639)*.

## 53.7 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 53.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart)



*Chart Properties, Presentation*

This tab is used collectively for bar charts, line charts, combo charts, radar charts and mekko charts.

The **Bar Settings** group contains various display options for bars that are used in bar charts and combo charts.

**Bar Distance (-6 - 8)** Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.

- Cluster Distance (0 - 8)** Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.
- Allow Thin Bars** For charts with a 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. Bars are by default drawn with a minimum width of four pixels, to make them clearly distinguishable. Check this option to allow compression of bars to a width of 1 pixel.
- Show All Bars** For charts with a 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. Check this option to force the drawing of all data points. Bars can be compressed (as for **Allow Thin Bars**) and some may also be partially obscured by others.

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 **Display Options** in *Chart Properties: Expressions (page 629)* page.

- Max Values Shown** In this box you can specify an upper limit for the number of 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.
- Vertical** Shows the values vertically.
- Plot Values Inside Segments** Marking this check box will plot values on data points inside the segments instead of on top of them.

In the **Error Bars** group display options for any error bars used in the chart is determined

- Width** Specifies the width of error bars.
- Thickness** Specifies the thickness of error bars.
- Color** Sets a color for error bars.

In the **Line/Symbol Settings** group display options for lines and data point symbols that are used in line charts and combo charts are determined. It is also possible to determine the width of trendlines.

- Line Width** Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).
- Symbol Size** Determines the size of symbols, if a symbol representation is specified.
- Trendline Width** This setting determines the width of trendlines.
- Use Full Symbol Set** This alternative makes more symbol representation available (rings, triangles etc.)
- Semi-transparent** Check this option if you want filled lines to be drawn semi-transparent.
- Pop-up Labels** Check this option to display the corresponding dimension value in a pop-up window when the mouse pointer touches a value.

- Highlight** With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.
- Suppress Zero-Values** This check box eliminates dimensions that are empty or contain only zeros. This option is selected by default.
- Zero on Bars**  
This option is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Points** is selected for the chart expression under **Display Options** in *Chart Properties: Expressions (page 629)*, 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.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

- Show Legend** Check this alternative to include a legend in the chart (checked by default). It is possible to change the *Legend Settings (page 646)* by clicking the **Settings...** button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.
- Limit Legend (Characters)** Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

- Enable X-Axis Scroll-bar** Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under **When Number of Items Exceeds**.
- Reversed** Checking the box displays the values in reversed order.

In the **Reference Lines** group you can define reference (grid) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

- Add** Opens the *Reference Lines (page 647)* dialog where you can create a new reference line in the chart.
- Edit** Highlight an existing reference line in the list and click this button to edit its properties in the *Reference Lines (page 647)* dialog.
- Delete** Highlight an existing reference line in the list and click this button to delete it from the list.

The **Text in Chart** group is used for adding free-floating text to the chart.

- Add** Opens the *Chart Text (page 648)* dialog, where you can create a new chart text.

<b>Edit</b>	Highlight an existing text in the list and click this button to edit its properties in the <i>Chart Text (page 648)</i> dialog.
<b>Delete</b>	Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in the layout edit mode. See *Sizing and Moving Chart Components (page 612)*.

## 53.9 Axes

On the **Axes** page you can set the display properties for the x- and y-axes.

This page is identical to *Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart) (page 649)*.

## 53.10 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 53.11 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 53.12 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 53.13 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 53.14 Caption

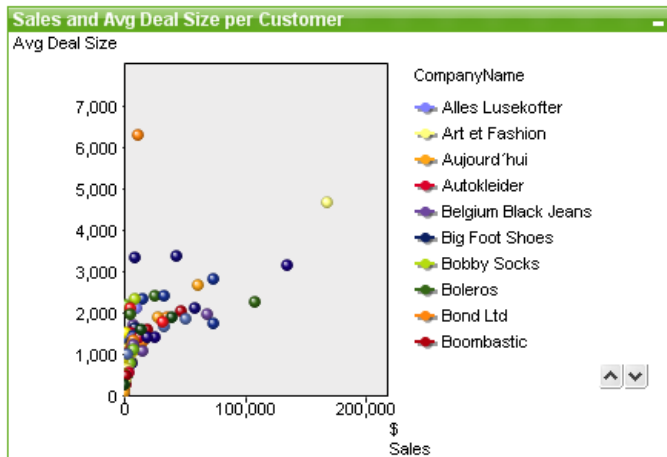
On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.






## 54 Scatter Chart



An example of a Scatter chart

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

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, for example, country (population and population growth).

The easiest way to create a new scatter chart is to click the **Create Chart**  button in the toolbar.

By a right-click on the scatter chart the *Scatter Chart: Object Menu* (page 681) will be displayed. It can also be accessed from the **Object** menu, when the scatter chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).




### 54.1 Scatter Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.

<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General</i> (page 91) dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.

<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Values</b> Copies the values to the clipboard in the form of a table. <b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 54.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

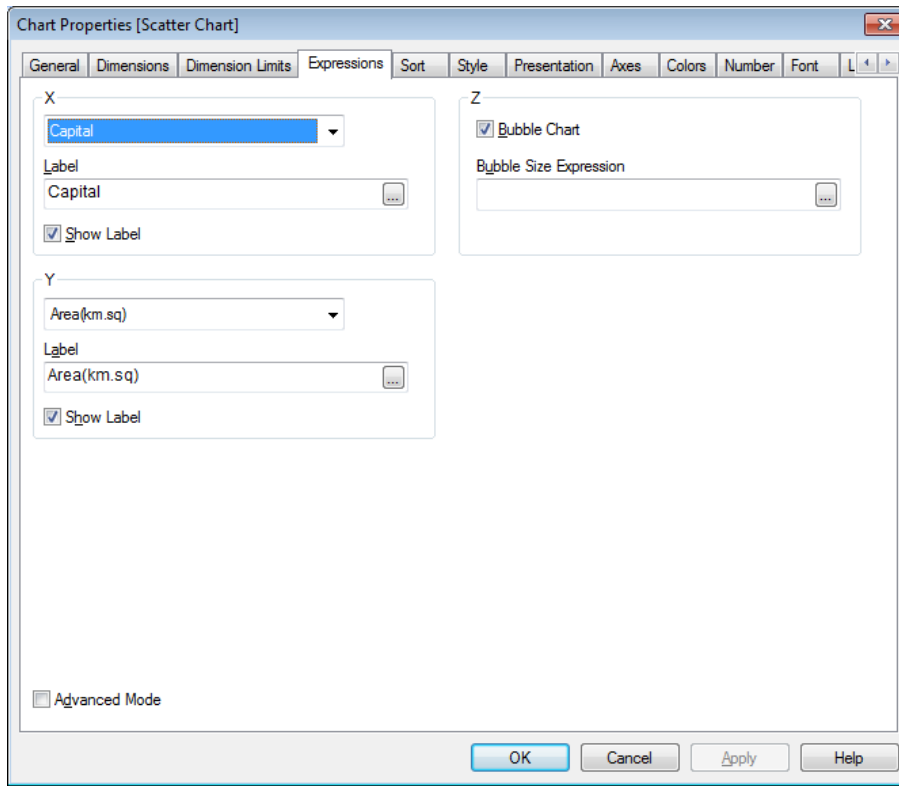
## 54.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 54.4 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 54.5 Chart Properties: Expressions (Scatter Chart)



*Chart Properties, Expressions (Scatter Chart)*

The **Chart Properties: Expressions** page is reached by a right-click on a scatter chart and selecting **Properties** from the **Object** menu. This is a simplified **Expressions** dialog that is only displayed when the alternative **Advanced Mode** is not checked.

The **X** group is used for determining the x-dimension. In the same manner, the y-dimension is created in the **Y** group. In this simplified **Expressions** dialog, all x- and y-dimensions will be defined as **Avg(FieldName)**. The Avg (average) is used as default, since it is the function most likely to produce a valid scatter.

For either dimension, a **Label** can be specified. This label can also be defined as a *Calculated Formula* (page 871). Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

In the **Z** group you can specify an optional expression which will be used to calculate the size of each data point plotted in the scatter:

**Bubble Chart** If this option is checked 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. Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

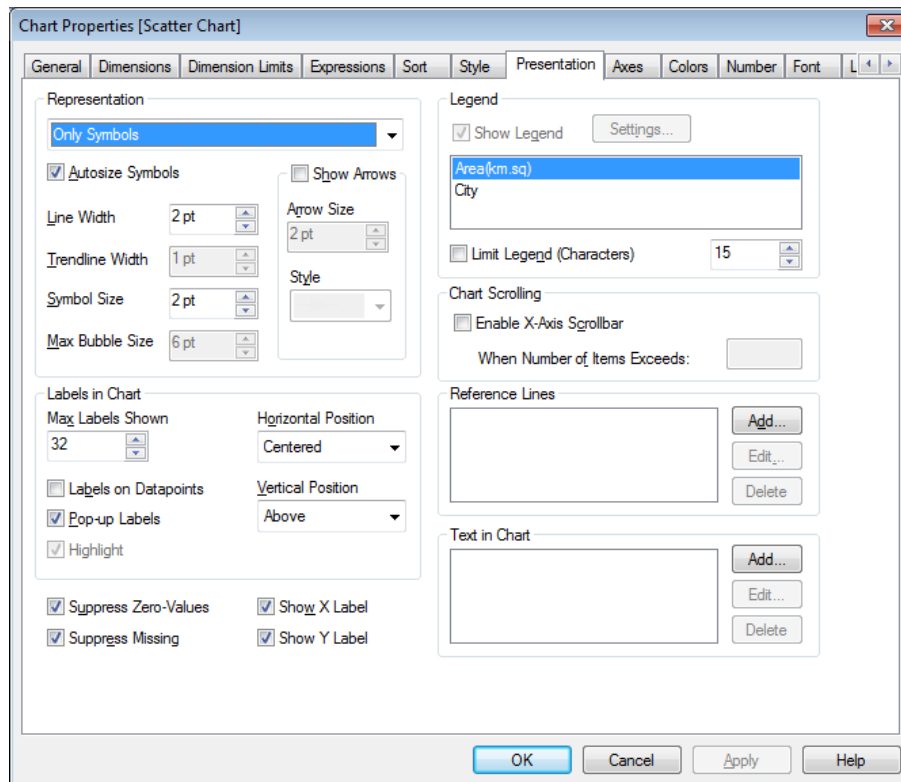
## 54.6 Sort

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. See *Chart Properties: Sort* (page 639).

## 54.7 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 54.8 Chart Properties: Presentation (Scatter- Grid Chart)



*Chart Properties, Presentation (Scatter and Grid Chart)*

This property page is used collectively for scatter charts and grid charts. When a scatter chart has two field dimensions instead of one defined on the **Dimensions** page, each value in the first field dimension will result in one scatter data point for each of its associated values in the second one. These may then be connected by lines. The following display options are available in the **Representation** group:

- Only Lines**                      Data is represented by lines between data points.
- Only Symbols**                      Data is represented by symbols, drawn at the data points.
- Both Lines and Symbols**                      A combination of the alternatives above.
- Autosize Symbols**                      Adjusts the maximum bubble size to the size of the scatter chart/to the number of values in the grid chart.
- Line Width**                      Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches ("), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).
- Trendline Width**                      This setting determines the width of trend lines.
- Symbol Size**                      Determines the size of symbols, if a symbol representation is specified.

<b>Max Bubble Size</b>	Sets the size of the largest bubble in the chart. Available only for some looks (selected on the <b>Style</b> tab) in combination with <b>Only Symbols</b> or <b>Both Line and Symbol</b> for <b>Representation</b> , and if <b>Autosize Symbols</b> is deselected.
<b>Show Arrows</b>	Set display options for arrows on connecting lines (See the <b>Representation</b> group, above!). Arrows will be directed between scatter points in the sequential order defined by the sort order of the second field dimension. Check this alternative to have arrow heads displayed on the connecting lines.
<b>Arrow Size</b>	The size of the arrow head can be edited here.
<b>Style</b>	Select from several styles from the drop-down list.

In the **Labels in Chart** group you may set display options for labels on data points. The labels plotted are the same as the ones in the legend.

<b>Max Labels Shown</b>	Limits the number of plotted labels. Setting this number too high may degrade chart clarity.
<b>Labels on Datapoints</b>	Check this alternative to have labels displayed.
<b>Pop-up Labels</b>	Check this alternative to display the corresponding dimension value in a pop-up window when pointing in the chart.
<b>Highlight</b>	With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.
<b>Horizontal Position</b>	Select the horizontal orientation from the drop-down list: <b>Left</b> , <b>Centered</b> or <b>Right</b> .
<b>Vertical Position</b>	Select the vertical orientation from the drop-down list: <b>Above</b> , <b>Centered</b> or <b>Below</b> .

The check boxes **Suppress Zero-Values** and **Suppress Missing** eliminate dimensions that are empty or contain only zeros from the table.

<b>Show X Label</b>	The label of x-dimension specified in the <b>X</b> group of <i>Chart Properties: Expressions (Scatter Chart) (page 684)</i> page is drawn at the end of the x-axis.
<b>Show Y Label</b>	The label of y-dimension specified in the <b>Y</b> group of <i>Chart Properties: Expressions (Scatter Chart) (page 684)</i> page is drawn at the end of the y-axis.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

<b>Show Legend</b>	Check this alternative to include a legend in the chart (checked by default). It is possible to change the <i>Legend Settings (page 646)</i> by clicking the <b>Settings...</b> button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.
<b>Limit Legend (Characters)</b>	Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

**Enable X-Axis Scroll-bar** Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under **When Number of Items Exceeds**.

In the **Reference Lines** group you can define reference (grid-) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

**Add** Opens the *Reference Lines (page 647)* dialog where a new reference line in the chart can be created.

**Edit** Highlight an existing reference line in the list and click this button to edit its properties in the *Reference Lines (page 647)* dialog.

**Delete** Highlight an existing reference line in the list and click this button to delete it from the list.

The **Texts in Chart** group is used for adding free-floating text to the chart.

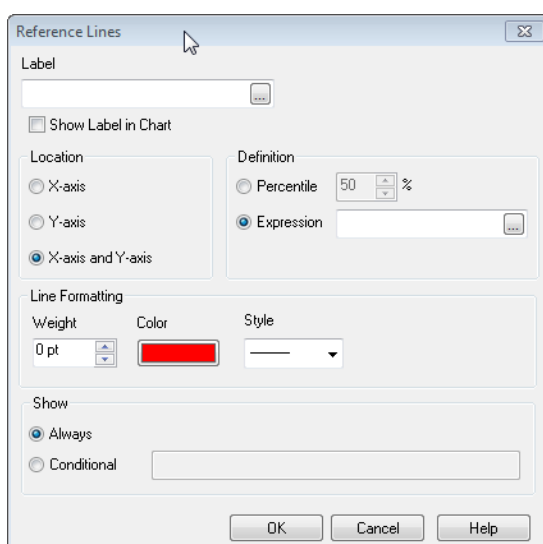
**Add** Opens the *Chart Text (page 648)* dialog where a new chart text in the chart can be created.

**Edit** Highlight an existing text in the list and click this button to edit its properties in the *Chart Text (page 648)* dialog.

**Delete** Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in chart layout edit mode, see *Sizing and Moving Chart Components (page 612)*.

## Reference Lines



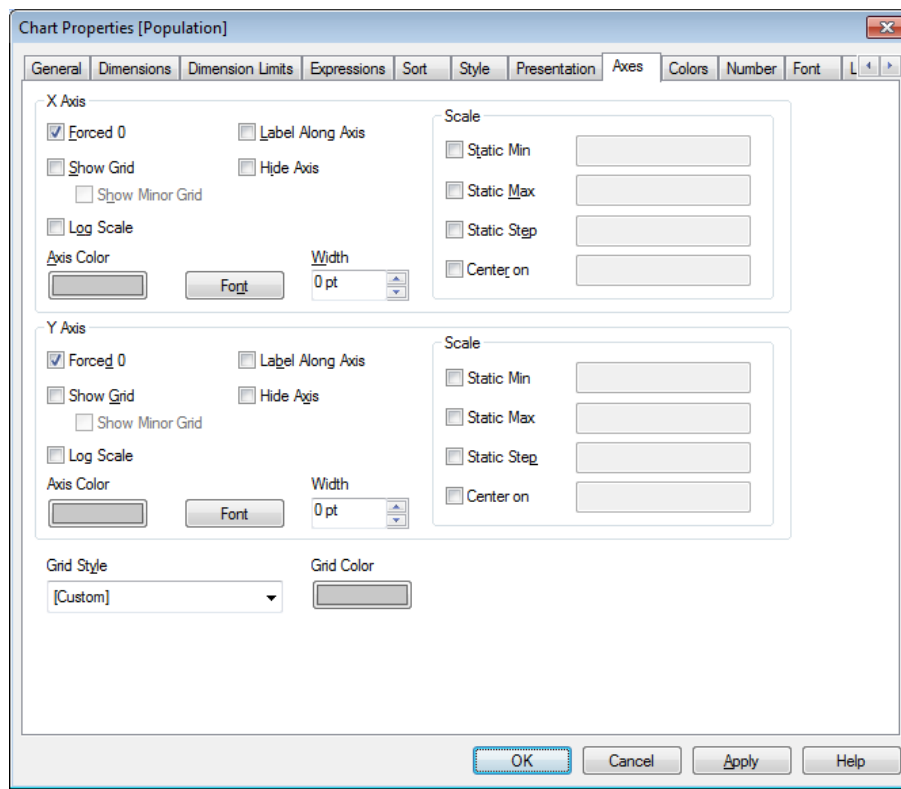
*The Reference Lines dialog for Scatter and Grid chart.*

The dialog may vary slightly in appearance depending on the kind of chart that is used. A reference line is a line intersecting the chart plot area from a given point on one or both axes. It may be used, for example, 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 originates.

<b>Label</b>	Enter a label to be drawn next to the reference line. The default value used is the expression. The label may be defined as a calculated expression.
<b>Show Label in Chart</b>	Enable this setting if the label should appear next to the reference line.
<b>Location</b>	Sets from which axis the reference line should originate, choose between <b>X-axis</b> , <b>Y-axis</b> and <b>X-axis and Y-axis</b>
<b>Definition</b>	Sets the value at which the reference line should be drawn. The value be either a fixed <b>Percentile</b> (enter a value between 1 and 100 in the edit box) of the current chart data or an arbitrary numeric <b>Expression</b> .
<b>Line Formatting</b>	Defines the layout of the reference line: <b>Weight</b> 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). <b>Color</b> Sets the color for the reference line. <b>Style</b> Specifies the style for the reference line, for example, continuous, dashed or dotted.
<b>Show</b>	Specifies the condition for displaying the reference line. <b>Always</b> The reference line will always be displayed. <b>Conditional</b> The reference line will be displayed 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.



## 54.9 Chart Properties: Axes (Scatter- Grid Chart)



*Chart Properties, Axes (Scatter and Grid Chart)*

This dialog is opened from the **Chart Properties** dialog of scatter charts and grid charts.

In this dialog the appearance of the x- and y-axes of the chart can be set.

The contents of the dialog:

### **X Axis**

- |                                    |   |
|------------------------------------|---|
| <b>Forced 0</b>                    | The y-axis will cross at $x = 0$ .  |
| <b>Show Grid / Show Minor Grid</b> | The scale ticks of the x-axis will be extended into grid lines that can be further formatted by the <b>Grid Style</b> and <b>Grid Color</b> settings. |
| <b>Log Scale</b>                   | A logarithmic scale will be used.   |
| <b>Label Along Axis</b>            | Changes the position of the x-axis label.   |
| <b>Hide Axis</b>                   | The x-axis will not be shown.   |

Set a separate **Axis Color** and **Width** for the x-axis as well as a separate **Font** for the scale numbering, by clicking the appropriate alternatives.

**Scale****Static Min**

The scale of the x-axis will not change with document state. Check this alternative to set a fixed minimum value for the x-axis in the edit box.

**Static Max**

Check this alternative to set a fixed maximum value for the x-axis.

**Static Step**

Check this alternative to set a fixed interval between scale ticks for the x-axis.

**Center on**

Check this alternative to set a fixed value on the x-axis which will be positioned in the center of the plot area. If the **Center on** value conflicts with **Static Min** and/or **Static Max** the **Center on** value will have priority.

Values entered in the **Scale** group may be specified as a *Calculated Formula* (page 871). Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

**Y Axis****Forced 0**

The x-axis will cross at  $y = 0$

**Show Grid / Show  
Minor Grid**

The scale ticks of the y-axis will be extended into grid lines that can be further formatted by the **Grid Style** and **Grid Color** settings.

**Log Scale**

A logarithmic scale will be used.

**Hide Axis**

The y-axis will not be shown.

**Label Along Axis**

Changes the position and orientation of the y-axis label.

Set a separate **Axis Color** and **Width** for the y-axis as well as a separate **Font** for the scale numbering, by clicking the appropriate alternatives.

**Scale****Static Min**

The scale of the y-axis will not change with document state. Check this alternative to set a fixed minimum value for the y-axis in the edit box.

**Static Max**

Check this alternative to set a fixed maximum value for the y-axis.

**Static Step**

Check this alternative to set a fixed interval between scale ticks for the y-axis.

**Center on**

Check this alternative to set a fixed value on the y-axis which will be positioned in the center of the plot area. If the **Center on** value conflicts with **Static Min** and/or **Static Max** the **Center on** value will have priority.

Values entered in the **Scale** group may be specified as a *Calculated Formula* (page 871). Click the ... button to open the **Edit Expression** dialog for easier editing of long formulas.

**Grid Style**

With **Show Grid** checked, you can choose from the available grid styles in the drop-down list.

**Grid Color**

This alternative lets you pick a color for the grid.

## 54.10 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors* (page 652).

## 54.11 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number* (page 655).

## 54.12 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font* (page 491).

## 54.13 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout* (page 492).

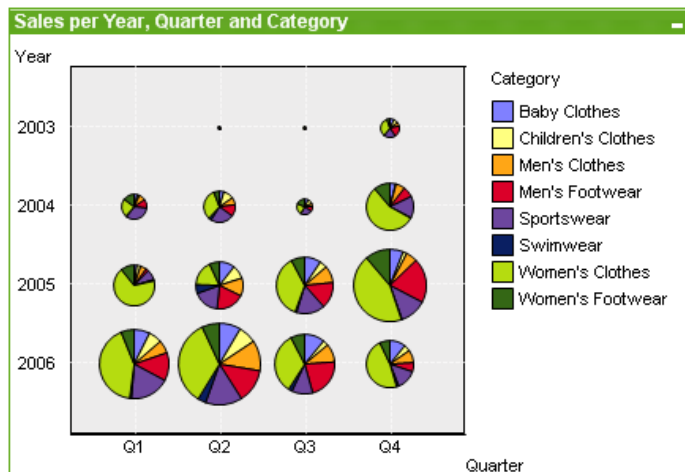
## 54.14 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption* (page 496).




## 55 Grid Chart



*An example of a grid chart*

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

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.

The easiest way to create a new grid chart is to click the **Create Chart**  button in the toolbar.

By a right-click on the grid chart the *Grid Chart: Object Menu* (page 693) will be displayed. It can also be accessed from the **Object** menu, when the grid chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).




### 55.1 Grid Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.

<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.

<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Values</b> Copies the values to the clipboard in the form of a table. <b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 55.2 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 55.3 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 55.4 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

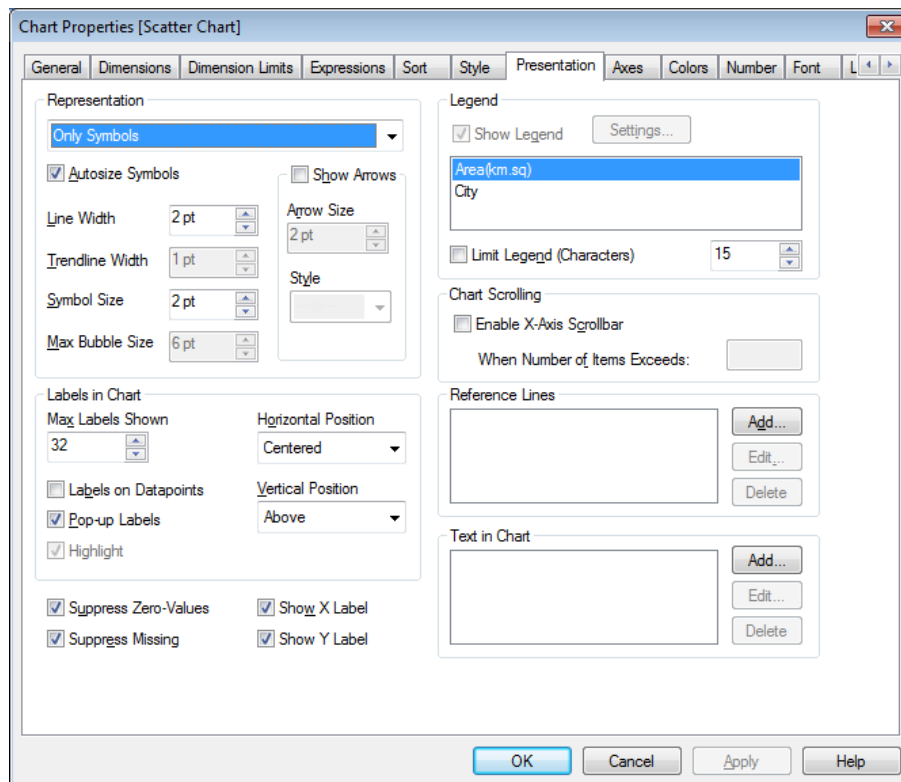
## 55.5 Sort

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. See *Chart Properties: Sort (page 639)*.

## 55.6 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 55.7 Chart Properties: Presentation (Scatter- Grid Chart)



*Chart Properties, Presentation (Scatter and Grid Chart)*

This property page is used collectively for scatter charts and grid charts. When a scatter chart has two field dimensions instead of one defined on the **Dimensions** page, each value in the first field dimension will result in one scatter data point for each of its associated values in the second one. These may then be connected by lines. The following display options are available in the **Representation** group:

- |                               |   |
|-------------------------------|---|
| <b>Only Lines</b>             | Data is represented by lines between data points.   |
| <b>Only Symbols</b>           | Data is represented by symbols, drawn at the data points.   |
| <b>Both Lines and Symbols</b> | A combination of the alternatives above.  |
| <b>Autosize Symbols</b>       | Adjusts the maximum bubble size to the size of the scatter chart/to the number of values in the grid chart. |



<b>Line Width</b>	Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).
<b>Trendline Width</b>	This setting determines the width of trend lines.
<b>Symbol Size</b>	Determines the size of symbols, if a symbol representation is specified.
<b>Max Bubble Size</b>	Sets the size of the largest bubble in the chart. Available only for some looks (selected on the <b>Style</b> tab) in combination with <b>Only Symbols</b> or <b>Both Line and Symbol</b> for <b>Representation</b> , and if <b>Autosize Symbols</b> is deselected.
<b>Show Arrows</b>	Set display options for arrows on connecting lines (See the <b>Representation</b> group, above!). Arrows will be directed between scatter points in the sequential order defined by the sort order of the second field dimension. Check this alternative to have arrow heads displayed on the connecting lines.
<b>Arrow Size</b>	The size of the arrow head can be edited here.
<b>Style</b>	Select from several styles from the drop-down list.

In the **Labels in Chart** group you may set display options for labels on data points. The labels plotted are the same as the ones in the legend.

<b>Max Labels Shown</b>	Limits the number of plotted labels. Setting this number too high may degrade chart clarity.
<b>Labels on Datapoints</b>	Check this alternative to have labels displayed.
<b>Pop-up Labels</b>	Check this alternative to display the corresponding dimension value in a pop-up window when pointing in the chart.
<b>Highlight</b>	With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.
<b>Horizontal Position</b>	Select the horizontal orientation from the drop-down list: <b>Left</b> , <b>Centered</b> or <b>Right</b> .
<b>Vertical Position</b>	Select the vertical orientation from the drop-down list: <b>Above</b> , <b>Centered</b> or <b>Below</b> .

The check boxes **Suppress Zero-Values** and **Suppress Missing** eliminate dimensions that are empty or contain only zeros from the table.

<b>Show X Label</b>	The label of x-dimension specified in the <b>X</b> group of <i>Chart Properties: Expressions (Scatter Chart)</i> (page 684) page is drawn at the end of the x-axis.
<b>Show Y Label</b>	The label of y-dimension specified in the <b>Y</b> group of <i>Chart Properties: Expressions (Scatter Chart)</i> (page 684) page is drawn at the end of the y-axis.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

- Show Legend** Check this alternative to include a legend in the chart (checked by default). It is possible to change the *Legend Settings (page 646)* by clicking the **Settings...** button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.
- Limit Legend (Characters)** Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

- Enable X-Axis Scrollbar** Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under **When Number of Items Exceeds**.

In the **Reference Lines** group you can define reference (grid-) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

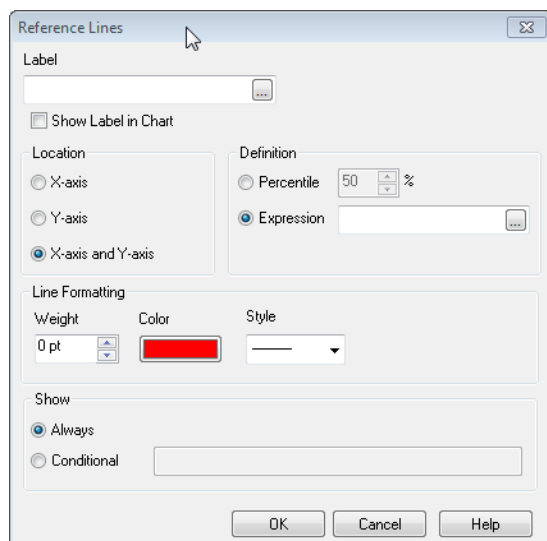
- Add** Opens the *Reference Lines (page 647)* dialog where a new reference line in the chart can be created.
- Edit** Highlight an existing reference line in the list and click this button to edit its properties in the *Reference Lines (page 647)* dialog.
- Delete** Highlight an existing reference line in the list and click this button to delete it from the list.

The **Texts in Chart** group is used for adding free-floating text to the chart.

- Add** Opens the *Chart Text (page 648)* dialog where a new chart text in the chart can be created.
- Edit** Highlight an existing text in the list and click this button to edit its properties in the *Chart Text (page 648)* dialog.
- Delete** Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in chart layout edit mode, see *Sizing and Moving Chart Components (page 612)*.

## Reference Lines



*The Reference Lines dialog for Scatter and Grid chart.*

The dialog may vary slightly in appearance depending on the kind of chart that is used. A reference line is a line intersecting the chart plot area from a given point on one or both axes. It may be used, for example, 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 originates.

- Label** Enter a label to be drawn next to the reference line. The default value used is the expression. The label may be defined as a calculated expression.
- Show Label in Chart** Enable this setting if the label should appear next to the reference line.
- Location** Sets from which axis the reference line should originate, choose between **X-axis**, **Y-axis** and **X-axis and Y-axis**
- Definition** Sets the value at which the reference line should be drawn. The value be either a fixed **Percentile** (enter a value between 1 and 100 in the edit box) of the current chart data or an arbitrary numeric **Expression**.
- Line Formatting** Defines the layout of the reference line:
- 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** Sets the color for the reference line.
- Style** Specifies the style for the reference line, for example, continuous, dashed or dotted.

**Show**

Specifies the condition for displaying the reference line.

**Always**

The reference line will always be displayed.

**Conditional**

The reference line will be displayed 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.

## 55.8 Axes

On the **Axes** page you can set the display properties for the x- and y-axes.

This page is identical to *Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart)* (page 649).

## 55.9 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors* (page 652).

## 55.10 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number* (page 655).

## 55.11 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font* (page 491).

## 55.12 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

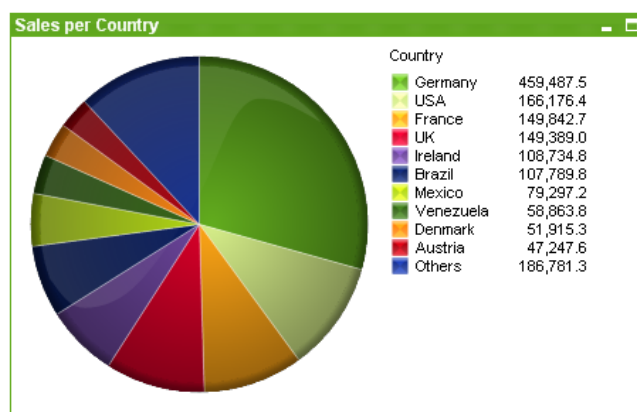
See *Layout* (page 492).

## 55.13 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption* (page 496).

## 56 Pie Chart



*An example of a pie chart*

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

Pie charts normally show the relation between a single dimension and a single expression, but can sometimes have two dimensions.

The quickest way to create a new pie chart is to select *Quick Chart Wizard* (page 767) from the **Tools** menu.

By a right-click on the pie chart the *Pie Chart: Object Menu* (page 701) will be displayed. It can also be accessed from the **Object** menu, when the pie chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

---

Note: The Pie Chart object is limited to displaying 500 slices.

---




### 56.1 Pie Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.
<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.

<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127. <b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet. <b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet. <b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127. <b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.
<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.

<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Values</b> Copies the values to the clipboard in the form of a table.</p> <p><b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	<p>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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Maximize</b>	<p>Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Restore</b>	<p>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.</p>
<b>Help</b>	<p>Opens QlikView help.</p>
<b>Remove</b>	<p>Removes the active sheet object from the sheet.</p>

## 56.2 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 56.3 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 56.4 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

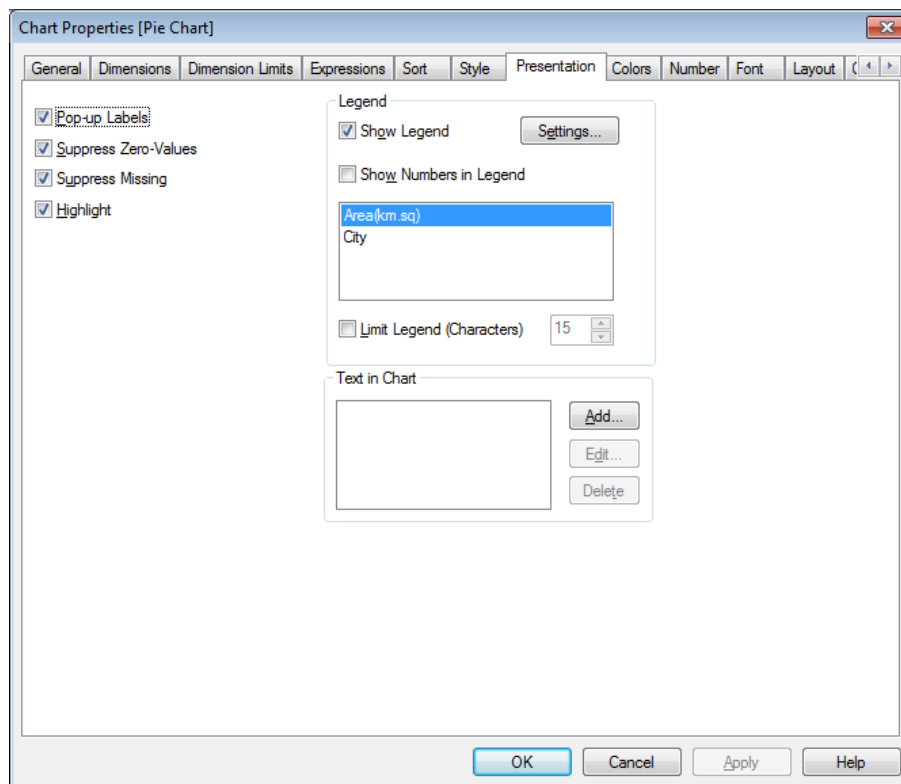
## 56.5 Sort

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. See *Chart Properties: Sort (page 639)*.

## 56.6 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 56.7 Chart Properties: Presentation (Pie Chart)



*Chart Properties, Presentation (Pie Chart)*

In this dialog you specify settings that determine how the pie-slices are displayed. The default values are:

- |                             |   |
|-----------------------------|---|
| <b>Pop-up Labels</b>        | Enable this setting to display the corresponding dimension value in a pop-up window when the mouse pointer touches the value.   |
| <b>Suppress Zero-Values</b> | If enabled, all combinations of the dimension fields returning zero or null from all expressions will be disregarded in the calculation. This option is enabled by default.   |
| <b>Suppress Missing</b>     | If enabled, 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 enabled by default. Turning it off can be useful only in special cases, for example, if you want to count null values in a chart. |



**Highlight** If enabled, hovering with the mouse over a pie segment will result in the segment being highlighted for better overview. The highlighting also applies in the legend where applicable.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

**Show Legend** Check this alternative to include a legend in the chart (checked by default). It is possible to change the *Legend Settings (page 646)* by clicking the **Settings...** button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.

**Show Numbers in Legend** If enabled, a legend showing numerical values will be included in the chart.

**Limit Legend (Characters)** Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

The **Texts in Chart** group is used for adding free-floating text to the chart.

**Add** Opens the *Chart Text (page 648)* dialog, where you can create a new chart text.

**Edit** Highlight an existing text in the list and click this button to edit its properties in the *Chart Text (page 648)* dialog.

**Delete** Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in chart layout edit mode, see *Sizing and Moving Chart Components (page 612)*.

## 56.8 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 56.9 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 56.10 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 56.11 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

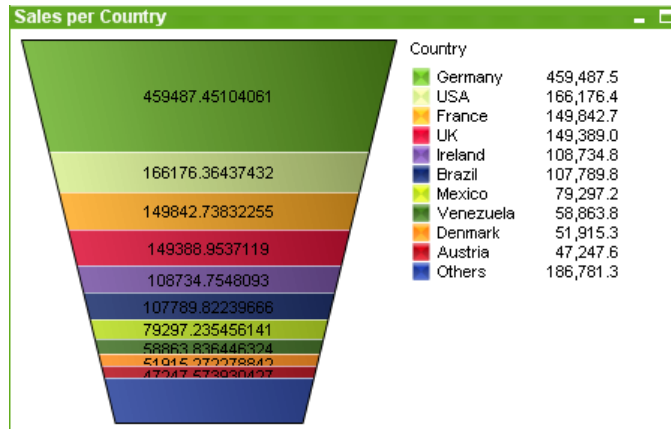
See *Layout (page 492)*.

## 56.12 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 57 Funnel Chart



An example of a Funnel chart

Charts are graphical representations of numerical data. The funnel chart is typically used for showing data in flows and processes. From a display standpoint it is related to the *Pie Chart* (page 701). 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 without regards to data points.

It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

By a right-click on the funnel chart the *Funnel Chart: Object Menu* (page 707) will be displayed. It can also be accessed from the **Object** menu, when the funnel chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).




### 57.1 Funnel Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

- |                      |   |
|----------------------|---|
| <b>Properties...</b> | Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.  |
| <b>Notes</b>         | Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.   |
| <b>Detach</b>        | The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached.<br>By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original. |
| <b>Attach</b>        | Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.  |

<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the fields used as dimensions in the chart.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. This command is only available if a QlikView printer is available on the system. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.
<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.

<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Values</b> Copies the values to the clipboard in the form of a table.</p> <p><b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	<p>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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Maximize</b>	<p>Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Restore</b>	<p>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.</p>
<b>Help</b>	<p>Opens QlikView help.</p>
<b>Remove</b>	<p>Removes the active sheet object from the sheet.</p>

## 57.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 57.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 57.4 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 57.5 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

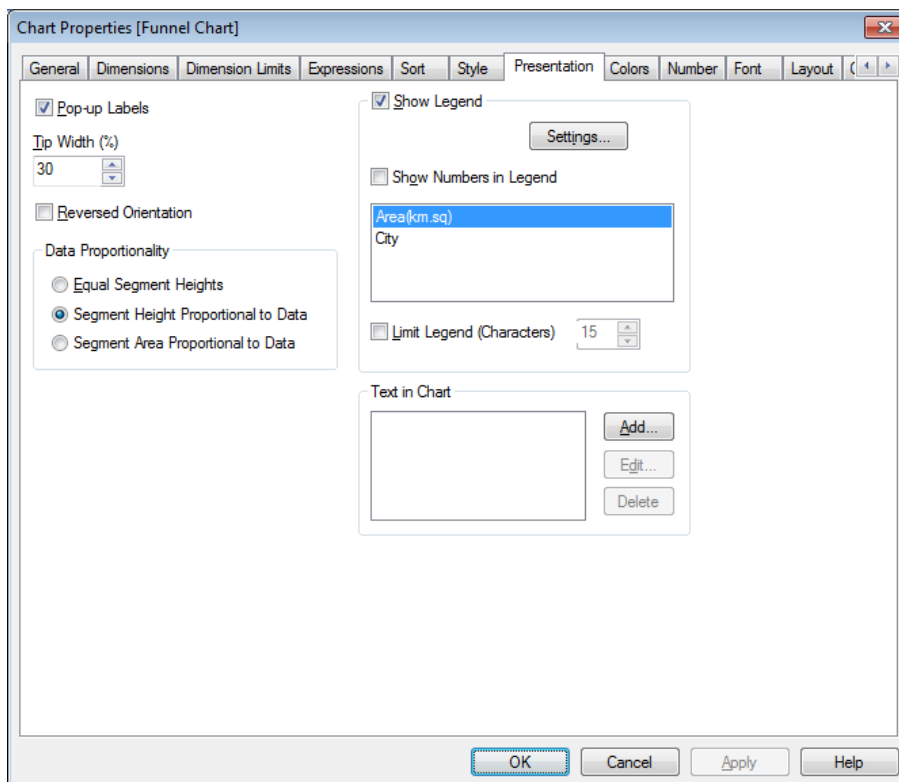
## 57.6 Sort

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. See *Chart Properties: Sort (page 639)*.

## 57.7 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 57.8 Chart Properties: Presentation (Funnel Chart)



*Chart Properties, Presentation (Funnel Chart)*

By changing the settings on this page, you modify the way the funnel chart is presented on the screen.

- |                      |   |
|----------------------|---|
| <b>Pop-up Labels</b> | Enable 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. |
| <b>Tip Width (%)</b> | Defines the width of the funnel tip as a percentage of the funnel mouth width.  |

<b>Reversed Orientation</b>	Enable this check box if you want to make the funnel point to the left or upwards
<b>Data Proportionality</b>	<p>This setting controls the degree to which the segments are proportional to the underlying data.</p> <p><b>Equal Segment Heights</b> 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.</p> <p><b>Segment Height Proportional to Data</b> Each segment is drawn with the a height (width for horizontal funnels) proportional to underlying data.</p> <p><b>Segment Area Proportional to Data</b> Each segment is drawn with the total area proportional to underlying data.</p>
<b>Show Legend</b>	<p>In the <b>Show Legend</b> group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.</p> <p><b>Settings...</b> Opens the <i>Legend Settings (page 646)</i> dialog where the legend settings can be modified.</p> <p><b>Show Numbers in Legend</b> Enable this check box to show values next to the dimension data labels.</p> <p><b>Limit Legend (Characters)</b> Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.</p>
<b>Text in Chart</b>	<p>In the <b>Text in Chart</b> group you can add free-floating text to the chart.</p> <p><b>Add</b> By pressing this button you open the <i>Chart Text (page 648)</i> dialog where you may create and edit free-floating texts to be displayed in the chart.</p> <p><b>Edit</b> Highlight an existing text in the list and click this button to edit its properties in the <i>Chart Text (page 648)</i> dialog. Double-clicking the text in the list has the same effect.</p> <p><b>Delete</b> Highlight an existing text in the list and click this button to delete it from the list.</p>

## 57.9 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 57.10 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 57.11 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 57.12 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

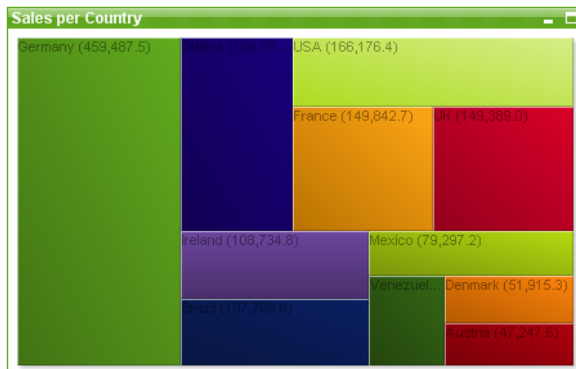
## 57.13 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.




## 58 Block Chart



An example of a Block chart

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

The block chart shows the relation between expression values as blocks of varying area. Up to three dimensions can be displayed, where each dimension is subdivided into 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.

The easiest way to create a new block chart is to click the **Create Chart**  button in the toolbar.

By right-clicking the block chart the *Block Chart: Object Menu* (page 713) will be displayed. It can also be accessed from the **Object** menu, when the block chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).




### 58.1 Block Chart: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.
<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.

<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. The maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. The minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.

<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Values</b> Copies the values to the clipboard in the form of a table. <b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page. <b>Object</b> 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.
<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 58.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 58.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 58.4 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

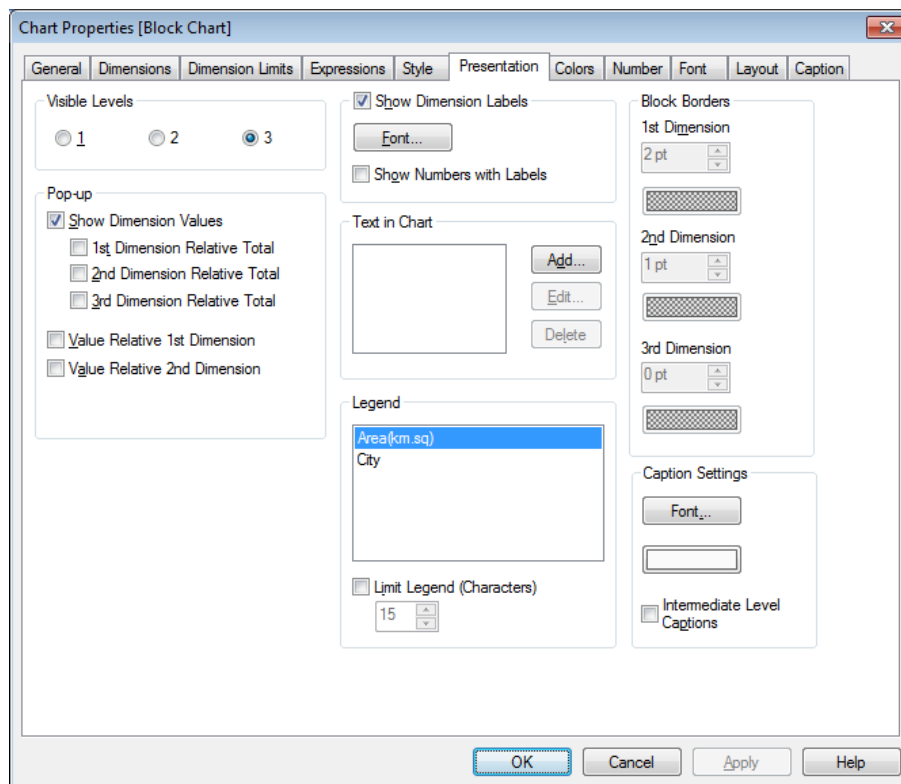
## 58.5 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

## 58.6 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 58.7 Chart Properties: Presentation (Block Chart)



*Chart Properties, Presentation (Block Chart)*

In this dialog you specify settings that determine how the data blocks are displayed. The default values are:

**Visible Levels** A block chart can display a maximum of three levels of blocks, equivalent to the three dimensions allowed. By selecting **1**, **2** or **3** you set the number of levels actually displayed.

By picking from the following options in the **Pop-up** group you control what information is shown in the pop-up that appears when you let the mouse pointer hovers over the data. (Note! These settings have no effect on expressions set to display as pop-up.)

**Show Dimension Values** With this option checked, all dimension values for a given data point are displayed. It is also required for the first three of the following settings.

<b>1st Dimension Relative Total</b>	With this option, the percentage of the indicated 1st dimension value relative to the total is included in the pop-up.
<b>2nd Dimension Relative Total</b>	The percentage of the indicated 2nd dimension value as defined within the indicated 1st dimension value relative to the total.
<b>3rd Dimension Relative Total</b>	As above, but for the 3rd dimension level.
<b>Value Relative 1st Dimension</b>	The percentage of the indicated block relative to the total of the indicated 1st dimension value.
<b>Value Relative 2nd Dimension</b>	The percentage of the indicated block relative to the total of the indicated 2nd dimension value.
<b>Show Dimension Labels</b>	Check this alternative to include labels in the chart (checked by default). It is also possible to change the <i>Font (page 491)</i> settings, by clicking the corresponding button. If <b>Show Numbers with Labels</b> is checked, numerical values will be included with the labels.

The **Text in Chart** group is used for adding free-floating text to the chart.

<b>Add</b>	Opens the <i>Chart Text (page 648)</i> dialog, where it is possible to create a new chart text.
<b>Edit</b>	Highlight an existing text in the list and click this button to edit its properties in the <i>Chart Text (page 648)</i> dialog.
<b>Delete</b>	Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in chart layout edit mode, see *Sizing and Moving Chart Components (page 612)*.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

<b>Show Legend</b>	Check this alternative to include a legend in the chart (checked by default). It is possible to change the <i>Legend Settings (page 646)</i> by clicking the <b>Settings...</b> button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.
<b>Limit Legend (Characters)</b>	Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

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.

For any dimension, click the spinner control to change the border width and the **Color** button to select a different border color.

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

<b>Font...</b>	Opens the <b>Font</b> dialog for setting a caption font.
<b>Color</b>	Opens the <b>Color Area</b> dialog for setting a caption base color.
<b>Intermediate Level Captions</b>	Mark this check box if you want captions drawn on the intermediate dimension level in three-dimensional block charts.

## 58.8 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 58.9 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 58.10 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 58.11 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

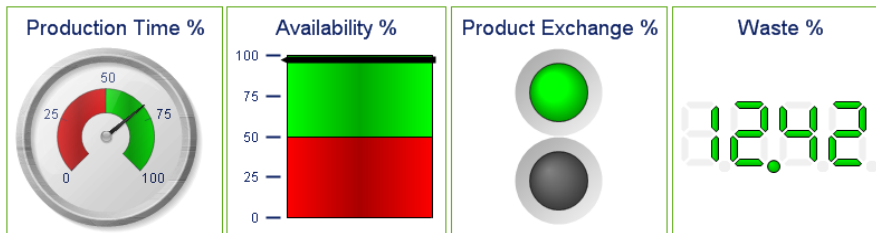
See *Layout (page 492)*.

## 58.12 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 59 Gauge Chart



*Some examples of Gauge charts*

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

Gauge charts are used to display the value of a single expression without dimensions.

The quickest way to create a new gauge chart is to select *Quick Chart Wizard* (page 767) from the **Tools** menu.

By a right-click on the gauge chart the *Gauge Chart: Object Menu* (page 719) will be displayed. It can also be accessed from the **Object** menu, when the gauge chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

### 59.1 Gauge Chart: Object Menu




Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached. By making a copy of a chart and detaching it, you may make direct comparisons between the copy and the original.
<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.

<b>Set Reference</b>	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 <b>Reference Mode</b> setting on the <b>Chart Properties: General</b> page. 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. The reference will be lost when closing the document or when reloading data. The maximum number of objects that can be included when using the Set Reference option is 500.
<b>Clear Reference</b>	This command is replaced with the <b>Set Reference</b> command when a reference is set. By choosing it the previously set reference will be cleared and the chart will revert to normal plot mode.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent of the chart) 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.
<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif.



<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Values</b> Copies the values to the clipboard in the form of a table.</p> <p><b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>
<b>Minimize</b>	<p>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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Maximize</b>	<p>Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.</p>
<b>Restore</b>	<p>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.</p>
<b>Help</b>	<p>Opens QlikView help.</p>
<b>Remove</b>	<p>Removes the active sheet object from the sheet.</p>

## 59.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 59.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 59.4 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

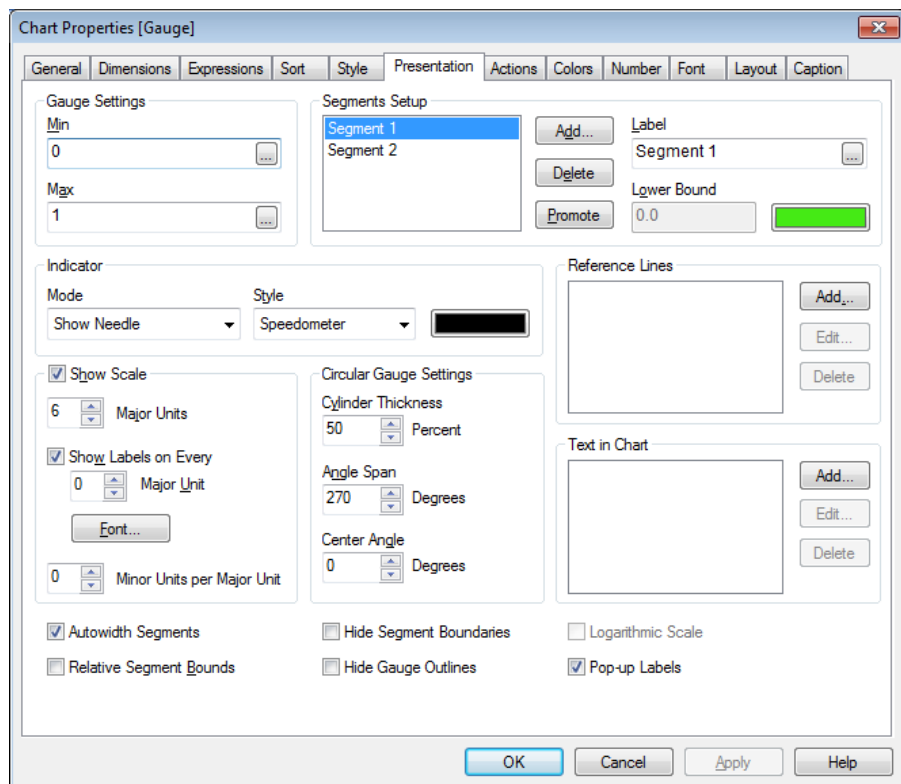
## 59.5 Sort

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. See *Chart Properties: Sort (page 639)*.

## 59.6 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 59.7 Chart Properties: Presentation (Gauge Chart)



*Chart Properties, Presentation (Circular Gauge Chart)*

The **Gauge Settings** group is used for determining the limits for the gauge. The selected gauge style determines what options become available in this dialog.

**Min** This is the minimum value limiting the range of the gauge. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*.

**Max** This is the maximum value limiting the range of the gauge. The value may be entered as a calculated formula. Click on the ... button to open the *Edit Expression Dialog (page 757)*.

The **Segments Setup** group is used for defining 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.

**Add** Click this button to add a new segment to the gauge.

<b>Delete</b>	Click this button to remove the currently selected segment from the gauge.
<b>Promote</b>	Use this button for sorting the segments in the list.
<b>Label</b>	Specifies a name for the selected segment. This name is for identification only and is not used when drawing the gauge chart.
<b>Lower Bound</b>	Specifies the gauge value where the selected segment starts. If the <b>Autowidth Segments</b> option is deselected (see below) this text box is enabled to edit the numerical value of the lower limit of the selected segment here. If the <b>Relative Segment Bounds</b> option is selected (see below) the segment bounds should be entered as numbers between 0 and 1, indicating a fraction of the total range between the <b>Min</b> and <b>Max</b> values.
<b>Color</b>	Assign a color to the selected segment by clicking the colored button. This opens the <i>Color Area (page 428)</i> dialog.

In the **Indicator** group you determine display options for the gauge indicator. This group is not available for LED style gauge charts.

<b>Mode</b>	Select one of the indication modes from the drop-down list.
<b>Style</b>	Select one of the gauge variants from the drop-down list.
<b>Show Scale</b>	The number entered for <b>nMajor Units</b> determines the main division of the scale. The labeling of the scale can be modified by the <b>Show Labels on Every nMajor Unit</b> setting and the <b>Font</b> button. The scale can be further detailed by the <b>nMinor Units per Major Unit</b> setting.

In the **Circular Gauge Settings** group, which is specific to circular style gauge charts, you specify the properties of the gauge shape.

<b>Cylinder Thickness</b>	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.
<b>Angle Span</b>	The angle in degrees between the Min and Max values in the gauge. Must be a value between 45 and 360.
<b>Center Angle</b>	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).

In the **Reference Lines** group you can define reference (grid-) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

<b>Add</b>	Opens the <i>Reference Lines (page 647)</i> dialog where you can create a new reference line in the chart.
<b>Edit</b>	Highlight an existing reference line in the list and click this button to edit its properties in the <i>Reference Lines (page 647)</i> dialog.
<b>Delete</b>	Highlight an existing reference line in the list and click this button to delete it from the list.

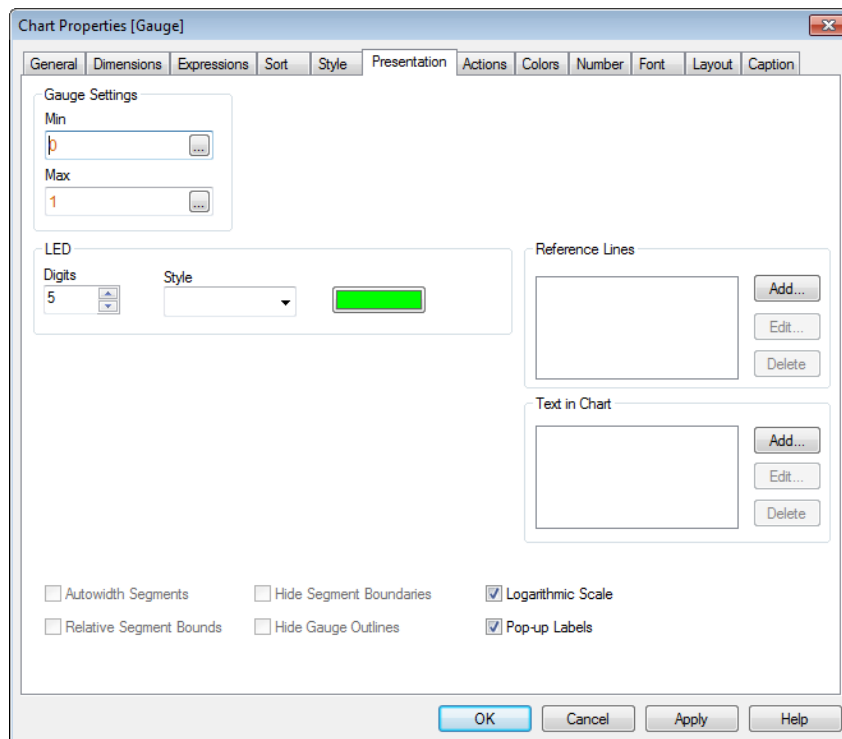
The **Text in Chart** group is used for adding free-floating text to the chart.

<b>Add</b>	Opens the <i>Chart Text (page 648)</i> dialog, where you can create a new chart text.
<b>Edit</b>	Highlight an existing text in the list and click this button to edit its properties in the <i>Chart Text (page 648)</i> dialog.
<b>Delete</b>	Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in chart layout edit mode, see *Sizing and Moving Chart Components (page 612)*.

<b>Autowidth Segments</b>	While this option is checked (default), the sizing of segments is calculated automatically, based on the gauge's <b>Min</b> value, <b>Max</b> value and the number of segments defined. If the option is deselected, the following options become available: <b>Lower Bound</b> , which is available from the <b>Segments Setup</b> group, <b>Relative Segment Bounds</b> and <b>Logarithmic Scale</b> .
<b>Relative Segment Bounds</b>	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 <b>Min</b> and <b>Max</b> values.
<b>Hide Segment Boundaries</b>	If this option is selected no outline will be drawn between the segment boundaries of circular and linear gauges, which is useful for creating background gradients with more than two colors.
<b>Hide Gauge Outlines</b>	If this option is selected no outline will be drawn around the gauge.
<b>Logarithmic Scale</b>	A logarithmic scale can be used, provided that all data points have positive values (>0).
<b>Pop-up Labels</b>	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.

In the **LED** group, which is specific to LED style gauge charts, you set the visual properties for the LED display.



*Chart Properties, Presentation (LED Gauge Chart)*

- Digits** Specifies the number of LED display digits shown.
- Color** Assign a color to the selected LED by clicking the colored button. This opens the *Color Area* (page 428) dialog.
- Style** Select one of the LED variants from the drop-down list.

## 59.8 Actions

In the **Actions** tab 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 *Actions* (page 552).

## 59.9 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors* (page 652).

## 59.10 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number* (page 655).

## 59.11 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font* (page 491).

## 59.12 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

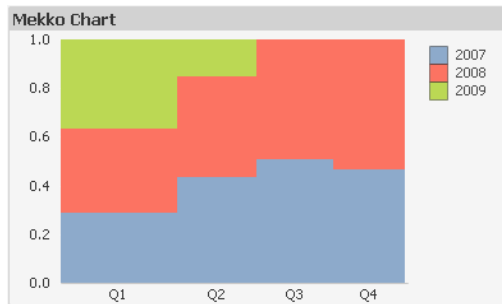
See *Layout (page 492)*.

## 59.13 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 60 Mekko Chart



*An example of a Mekko chart*

Charts are graphical representations of numerical data. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

Mekko charts present data using variable width bars. They can display up to three levels of data in a two-dimensional chart. Mekko charts are useful in such areas as market analysis.

By a right-click on the chart the *Mekko Chart: Object Menu* (page 727) will be displayed. It can also be accessed from the **Object** menu, when the chart is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

### 60.1 Mekko Chart: Object Menu

Mekko charts present data using variable width bars. They can display up to three levels of data in a two-dimensional chart. Mekko charts are useful in such areas as market analysis.

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.




The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the chart can be set. This command can also be invoked by the following keyboard shortcut: Alt + Enter.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments</i> (page 472) for further information.
<b>Detach</b>	The chart title is appended with the text "(Detached)" and the chart is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the chart is attached.
<b>Attach</b>	Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
<b>Clone</b>	Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.

<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu</i> (page 52) is activated or when the <i>Always Show Design Menu Items</i> (page 78) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the chart.
<b>Print...</b>	Opens the <i>Print: General</i> (page 91) dialog from which you can print the chart.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button 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 Qlik-Tech's homepage and installed separately.
<b>Send Values to Excel</b>	Exports the underlying data (the straight table equivalent to the chart) 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.
<b>Export...</b>	Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png.
<b>Copy to Clipboard</b>	<p>This menu contains the various copy options for the chart.</p> <p><b>Values</b> Copies the values to the clipboard in the form of a table.</p> <p><b>Image</b> Copies an image of the chart object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the <b>User Preferences: Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>



---

<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 60.2 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 60.3 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

## 60.4 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 60.5 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

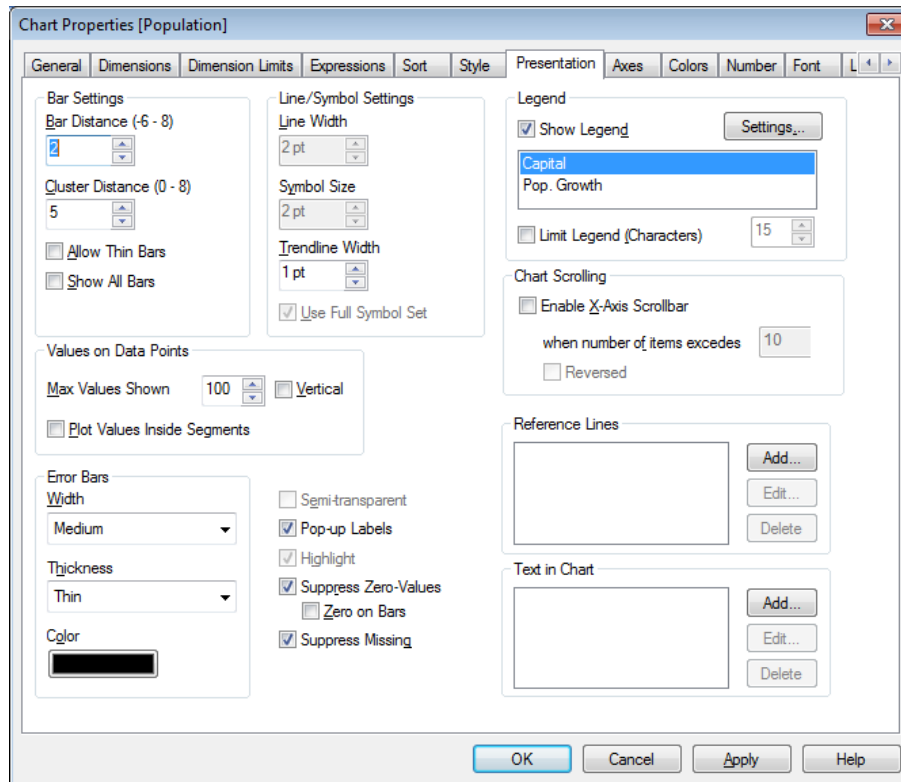
## 60.6 Sort

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. See *Chart Properties: Sort (page 639)*.

## 60.7 Style

In this page you can sort the values of the dimensions according to one of the available sort orders. This page is identical to *Chart Properties: Style (page 640)*.

## 60.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart)



*Chart Properties, Presentation*

This tab is used collectively for bar charts, line charts, combo charts, radar charts and mekko charts.

The **Bar Settings** group contains various display options for bars that are used in bar charts and combo charts.

- Bar Distance (-6 - 8)** Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.
- Cluster Distance (0 - 8)** Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.
- Allow Thin Bars** For charts with a 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. Bars are by default drawn with a minimum width of four pixels, to make them clearly distinguishable. Check this option to allow compression of bars to a width of 1 pixel.
- Show All Bars** For charts with a 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. Check this option to force the drawing of all data points. Bars can be compressed (as for **Allow Thin Bars**) and some may also be partially obscured by others.

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 **Display Options** in *Chart Properties: Expressions (page 629)* page.

**Max Values Shown** In this box you can specify an upper limit for the number of 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.

**Vertical** Shows the values vertically.

**Plot Values Inside Segments** Marking this check box will plot values on data points inside the segments instead of on top of them.

In the **Error Bars** group display options for any error bars used in the chart is determined

**Width** Specifies the width of error bars.

**Thickness** Specifies the thickness of error bars.

**Color** Sets a color for error bars.

In the **Line/Symbol Settings** group display options for lines and data point symbols that are used in line charts and combo charts are determined. It is also possible to determine the width of trendlines.

**Line Width** Determines the width of the line, if a line representation is specified. The value can be specified in mm, cm, inches (" , inch), pixels (px, pxl, pixel), points (pt, pts, point) or docunits (du, docunit).

**Symbol Size** Determines the size of symbols, if a symbol representation is specified.

**Trendline Width** This setting determines the width of trendlines.

**Use Full Symbol Set** This alternative makes more symbol representation available (rings, triangles etc.)

**Semi-transparent** Check this option if you want filled lines to be drawn semi-transparent.

**Pop-up Labels** Check this option to display the corresponding dimension value in a pop-up window when the mouse pointer touches a value.

**Highlight** With this option checked, symbols and/or lines are highlighted when the mouse pointer hovers over them. Where a legend is included in the chart, the highlighting applies here as well, making it possible to single out one of several overlapping values.

**Suppress Zero-Values** This check box eliminates dimensions that are empty or contain only zeros. This option is selected by default.

**Zero on Bars**

This option is only applicable when **Suppress Zero-Values** is deselected. If the check box is marked and **Values on Data Points** is selected for the chart expression under **Display Options** in *Chart Properties: Expressions (page 629)* , 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.

In the **Legend** group you can control the display of dimension data labels in the chart. Mark the check box to display data labels. Data labels are shown only for the current top level of the chart.

**Show Legend** Check this alternative to include a legend in the chart (checked by default). It is possible to change the *Legend Settings (page 646)* by clicking the **Settings...** button. If the chart is dimensionless, but has several expressions, unmarking this check box will show the expressions on the axis instead.

**Limit Legend (Characters)** Enable this check box to limit the length of the dimension value strings that are displayed on axes and in the chart legend. Truncated values will be followed by ... in the chart.

In the **Chart Scrolling** group you can make settings for scrolling in the chart.

**Enable X-Axis Scrollbar** Enable this check box to show a scroll control in place of the X-axis. The scroll bar can be used to scroll the selection of the X-axis values that are displayed. The number of values shown at any one time will be the number set under **When Number of Items Exceeds**.

**Reversed** Checking the box displays the values in reversed order.

In the **Reference Lines** group you can define reference (grid) lines intersecting the chart plot area from a given point on a continuous x-axis or a y-axis. Existing reference lines are listed in the window.

**Add** Opens the *Reference Lines (page 647)* dialog where you can create a new reference line in the chart.

**Edit** Highlight an existing reference line in the list and click this button to edit its properties in the *Reference Lines (page 647)* dialog.

**Delete** Highlight an existing reference line in the list and click this button to delete it from the list.

The **Text in Chart** group is used for adding free-floating text to the chart.

**Add** Opens the *Chart Text (page 648)* dialog, where you can create a new chart text.

**Edit** Highlight an existing text in the list and click this button to edit its properties in the *Chart Text (page 648)* dialog.

**Delete** Highlight an existing text in the list and click this button to delete it from the list.

Free-floating texts appear at the top left position in the chart, but can be repositioned when the chart is in the layout edit mode. See *Sizing and Moving Chart Components (page 612)*.

## 60.9 Axes

On the **Axes** page you can set the display properties for the x- and y-axes.

This page is identical to *Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart) (page 649)*.

## 60.10 Colors

On the **Colors** page you can set the display colors. This page is identical to *Chart Properties: Colors (page 652)*.

## 60.11 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 60.12 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 60.13 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 60.14 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.



# 61 Pivot Table

Country	Salesman	Year	Sales
Australia	Rolf Wesenlund	2005	1,030
		2006	1,210
	<b>Total</b>		<b>2,240</b>
<b>Total</b>			<b>2,240</b>
Azerbaijan			5,329
Bahrain			1,090
Bangladesh			4,240
Belarus			26,065
Belgium	Charles Ingvar Jönsson	2006	1,210
		2008	3,159
		2009	3,690
	<b>Total</b>		<b>8,059</b>
Belgium	John Cleaves	2008	2,550
		<b>Total</b>	<b>2,550</b>
	Tony Cedholt	2008	2,500
2009		4,249	
<b>Total</b>			<b>6,749</b>
<b>Total</b>			<b>17,358</b>

An example of a Pivot table

Charts are graphical representations of numerical data. The pivot table and the straight table are special cases as they display the data in table form while retaining all the properties of a chart. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

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.

The quickest way to create a new pivot table is to select *Quick Chart Wizard* (page 767) from the **Tools** menu.

By a right-click on the pivot table the *Pivot Table: Object Menu* (page 738) will be displayed. It can also be accessed from the **Object** menu, when the pivot table is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

## 61.1 Using the Pivot Table

In a pivot table dimensions (fields and expressions) can be shown on one vertical and one horizontal axis. The dimensions may be moved freely between or within the axes. This process is called "pivoting". In QlikView, pivoting is done by dragging and dropping with a mouse. Simply point the mouse somewhere in the field, click and drag it to the desired position. To make this easier, while a field is being moved, its borders are highlighted in blue.

### Note!

Pivoting is disabled if the **Allow Pivoting** check box in the *Chart Properties: Presentation (Pivot Table)* (page 742) is unchecked.

In QlikView, multi-dimensional pivot tables will display small + and - icons in its variable fields. A + icon indicates that the table can be further expanded for detail by revealing further variables, while a - icon indicates that it can be collapsed, which sacrifices detail for clarity or space.

From the *Pivot Table: Object Menu* (page 738) you also have access to the commands **Expand all**, **Collapse all**, **Collapse Dimension Rows** and **Collapse Dimension Columns** which effect these setting accordingly.

### Note!

If you use cyclic dimension groups, it is recommended that you always first **Expand all** and then expand or

collapse at individual dimension level to ensure that all levels, even levels not displayed, are properly expanded.

**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
		<b>Total</b>	<b>5</b>	<b>2.5</b>
	B	1998	1	1.0
		1999	1	1.0
		<b>Total</b>	<b>2</b>	<b>1.0</b>
<b>Total</b>			<b>7</b>	<b>1.8</b>
XYZ	A	1998	5	5.0
		1999	4	4.0
		<b>Total</b>	<b>9</b>	<b>4.5</b>
	B	1998	7	7.0
		1999	6	6.0
		<b>Total</b>	<b>13</b>	<b>6.5</b>
<b>Total</b>			<b>22</b>	<b>5.5</b>
<b>Total</b>			<b>29</b>	<b>3.6</b>

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	<b>4</b>	<b>2.0</b>
	1999	2	2.0	1	1.0	<b>3</b>	<b>1.5</b>
	<b>Total</b>	<b>5</b>	<b>2.5</b>	<b>2</b>	<b>1.0</b>	<b>7</b>	<b>1.8</b>
XYZ	1998	5	5.0	7	7.0	<b>12</b>	<b>6.0</b>
	1999	4	4.0	6	6.0	<b>10</b>	<b>5.0</b>
	<b>Total</b>	<b>9</b>	<b>4.5</b>	<b>13</b>	<b>6.5</b>	<b>22</b>	<b>5.5</b>
<b>Total</b>		<b>14</b>	<b>3.5</b>	<b>15</b>	<b>3.8</b>	<b>29</b>	<b>3.6</b>

Not only the dimension fields, but also the expression row can be pivoted:

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	<b>4</b>	<b>2.0</b>
	1999	2	2.0	1	1.0	<b>3</b>	<b>1.5</b>
	<b>Total</b>	<b>5</b>	<b>2.5</b>	<b>2</b>	<b>1.0</b>	<b>7</b>	<b>1.8</b>
XYZ	1998	5	5.0	7	7.0	<b>12</b>	<b>6.0</b>
	1999	4	4.0	6	6.0	<b>10</b>	<b>5.0</b>
	<b>Total</b>	<b>9</b>	<b>4.5</b>	<b>13</b>	<b>6.5</b>	<b>22</b>	<b>5.5</b>
<b>Total</b>		<b>14</b>	<b>3.5</b>	<b>15</b>	<b>3.8</b>	<b>29</b>	<b>3.6</b>

The result:

sum(Sales)				
Company	Year	Product	A	B
ABC	1998	sum(Sales)	3	1
		avg(Sales)	3.0	1.0
	1999	sum(Sales)	2	1
		avg(Sales)	2.0	1.0
	Total	sum(Sales)	5	2
		avg(Sales)	2.5	1.0
XYZ	1998	sum(Sales)	5	7
		avg(Sales)	5.0	7.0
	1999	sum(Sales)	4	6
		avg(Sales)	4.0	6.0
	Total	sum(Sales)	9	13
		avg(Sales)	4.5	6.5
<b>Total</b>		sum(Sales)	<b>14</b>	<b>15</b>
		avg(Sales)	<b>3.5</b>	<b>3.8</b>



## 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!

It is possible to disable the expand and collapse options for a pivot table by selecting the check box **Always Fully Expanded** on the **Chart Properties: Presentation** page of the pivot table.

### 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)		
Company	sum(Sales)	avg(Sales)
ABC	7	1.8
XYZ	22	5.5
<b>Total</b>	<b>29</b>	<b>3.6</b>

A click on the first + icon 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
	<b>Total</b>	<b>7</b>	<b>1.8</b>
XYZ		22	5.5
<b>Total</b>		<b>29</b>	<b>3.6</b>

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
	<b>Total</b>	<b>7</b>	<b>1.8</b>
XYZ	A	9	4.5
	B	13	6.5
	<b>Total</b>	<b>22</b>	<b>5.5</b>
<b>Total</b>		<b>29</b>	<b>3.6</b>

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 as follows:

sum(Sales)				
Company	Product	Year	sum(Sales)	avg(Sales)
ABC	A	1998	3	3.0
		1999	2	2.0
		<b>Total</b>	<b>5</b>	<b>2.5</b>
	B	1998	1	1.0
		1999	1	1.0
		<b>Total</b>	<b>2</b>	<b>1.0</b>
<b>Total</b>		<b>7</b>	<b>1.8</b>	
XYZ	A	1998	5	5.0
		1999	4	4.0
		<b>Total</b>	<b>9</b>	<b>4.5</b>
	B	1998	7	7.0
		1999	6	6.0
		<b>Total</b>	<b>13</b>	<b>6.5</b>
<b>Total</b>		<b>22</b>	<b>5.5</b>	
<b>Total</b>		<b>29</b>	<b>3.6</b>	

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 as follows:

sum(Sales)				
Company	Product	Year	sum(Sales)	avg(Sales)
ABC	A		5	2.5
	B	1998	1	1.0
		1999	1	1.0
		<b>Total</b>	<b>2</b>	<b>1.0</b>
<b>Total</b>			<b>7</b>	<b>1.8</b>
XYZ	A		9	4.5
	B	1998	7	7.0
		1999	6	6.0
		<b>Total</b>	<b>13</b>	<b>6.5</b>
<b>Total</b>			<b>22</b>	<b>5.5</b>
<b>Total</b>			<b>29</b>	<b>3.6</b>

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)							
Product		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
	<b>Total</b>	<b>5</b>	<b>2.5</b>	<b>2</b>	<b>1.0</b>	<b>7</b>	<b>1.8</b>
XYZ	1998	5	5.0	7	7.0	12	6.0
	1999	4	4.0	6	6.0	10	5.0
	<b>Total</b>	<b>9</b>	<b>4.5</b>	<b>13</b>	<b>6.5</b>	<b>22</b>	<b>5.5</b>
<b>Total</b>		<b>14</b>	<b>3.5</b>	<b>15</b>	<b>3.8</b>	<b>29</b>	<b>3.6</b>

## 61.2 Pivot Table: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.

The following commands are available (the commands may vary depending on what field you click on):

**Properties...** Opens the **Properties** dialog where the parameters defining the chart can be set. This command can also be invoked by the following keyboard shortcut: Alt+Enter.

**Notes** Allows creating and sharing notes about the current object. See *Notes and Comments* (page 472) for further information.




---

<p><b>Expand all</b>  <b>Collapse all</b>  <b>Collapse Dimension Rows</b>  <b>Collapse Dimension Columns</b></p>	<p>These commands operate on the expand (+) and collapse (-) icons that are displayed in multi-dimensional pivot tables.</p>
<b>Detach</b>	<p>The chart title is appended with the text "(Detached)" and the table is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the table is attached.</p> <p>By making a copy of a pivot table and detaching it, you may make direct comparisons between the copy and the original.</p>
<b>Attach</b>	<p>Attaches a detached pivot table. The pivot table becomes dynamically linked to the data. The command is available only if the pivot table is detached.</p>
<b>Clone</b>	<p>Makes an identical copy of the pivot table. If a detached pivot table is cloned the clone will be attached.</p>
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b>  Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b>  Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b>  Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b>  Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>
<b>Fit Columns to Data</b>	<p>Adjusts the width of each table column to its longest data string. Headers are included in the calculations.</p>
<b>Equal Column Width</b>	<p>If the mouse pointer is placed over a column in the pivot table this command becomes available in the float menu (not in the main menu bar <b>Object</b> menu). The command sets column width for all columns of the table to that of the column pointed at.</p> <p>Column width can be individually adjusted by moving the pointer to the right edge of the column (pointer changes appearance) and dragging.</p>
<b>Custom Format Cell</b>	<p>Opens the <i>Custom Format Cell Dialog (page 524)</i> which lets you format cells in the column and stripe which you clicked upon. This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated.</p>
<b>Change Value</b>	<p>Only available for expression columns containing an inputsum aggregation of an input field. Sets the cell clicked on in input edit mode. Equivalent to clicking the input icon in the cell.</p>

---

<b>Restore Values</b>	<p>Only available for expression columns containing an inputsum aggregation of an input field. Opens a cascade menu with three options.</p> <p><b>Restore Single Value</b> Restores the field values underlying the cell clicked upon on to their default values from the script.</p> <p><b>Restore Possible Values</b> Restores the values of all possible underlying field values to their default values from the script.</p> <p><b>Restore All Values</b> Restores the values of all underlying field values to their default values from the script.</p>
<b>Clear All Selections</b>	<p>Clears all selections in the dimensions and expressions of the table.</p>
<b>Print...</b>	<p>Opens the <i>Print: General (page 91)</i> dialog from which you can print the pivot table.</p>
<b>Print as PDF...</b>	<p>Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After pressing the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.</p>
<b>Send to Excel</b>	<p>Exports the table to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. 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.</p>
<b>Export...</b>	<p>Opens the <b>Save as</b> dialog where path, file name and file type for the exported table content can specified.</p> <p>The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files).</p>
<b>Copy to Clipboard</b>	<p>This cascade menu contains the various copy options for the sheet object.</p> <p><b>Full Table</b> Copies the table to the clipboard, complete with header and selection status.</p> <p><b>Table Data Area</b> Copies only the values of the table to the clipboard.</p> <p><b>Cell Value</b> Copies the text value of the cell right-clicked upon (when invoking the Object menu) to the clipboard.</p> <p><b>Image</b> 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 <b>User Preferences</b> dialog, <b>Export</b> page.</p> <p><b>Object</b> 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.</p>
<b>Linked Objects</b>	<p>Opens a menu with the following commands for linked objects.</p> <p><b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted.</p> <p><b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.</p>

---

<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 61.3 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 61.4 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

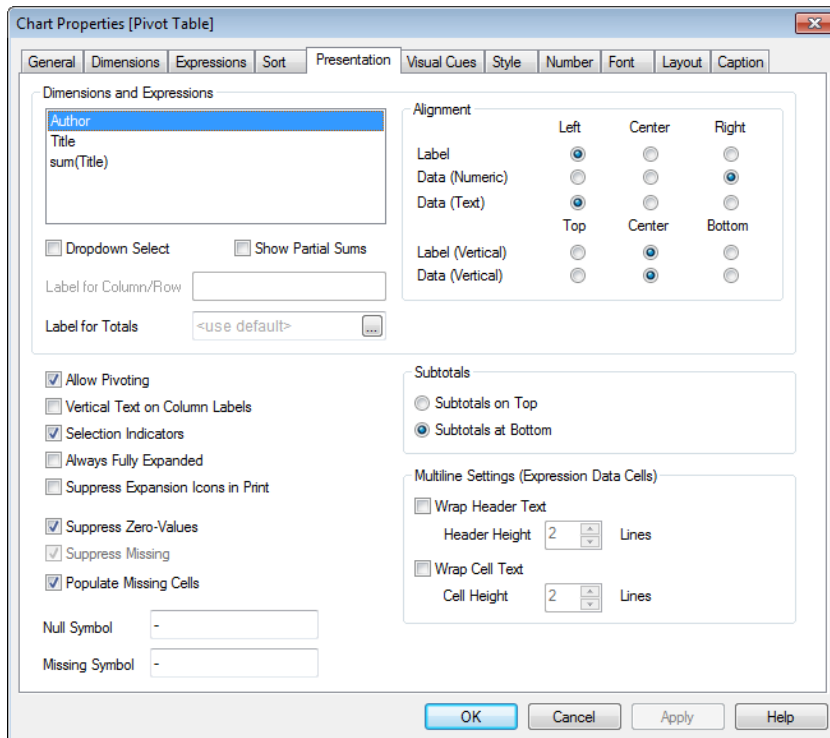
## 61.5 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

## 61.6 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. See *Chart Properties: Sort (page 639)*.

## 61.7 Chart Properties: Presentation (Pivot Table)



*Chart Properties, Presentation (Pivot Table)*

In the **Dimensions and Expressions** group all the field-dimensions and expressions of the pivot table are listed. Select one from the list in order to make individual adjustments to it.

### **Dropdown Select**

If enabled for a field column, a drop-down icon will appear to the right in the column header. By clicking the icon, a list box displaying all field values of the field will be opened over the table. Selections and searches may then be made 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**, **Center** or **Right**. When multi line cells and labels are used, **Label (Vertical)** and **Data (Vertical)** can be set to **Top**, **Center** or **Bottom**.

### **Allow Pivoting**

If this option is de-selected, the usual pivoting function of the pivot table will be disabled.

---

<b>Vertical Text on Column Labels</b>	Text for the column headers will be rotated to vertical.
<b>Selection Indicators</b>	With this option checked, a colored Indicator is displayed in the header of any field dimension where a selection has been made.
<b>Always Fully Expanded</b>	This alternative means that you will not be able to collapse dimensions by clicking on the - icons.
<b>Suppress Expansion Icons in Print</b>	Select this check box if you don't want the + and - icons for partial expand and collapse to be visible when printing the pivot table.
<b>Suppress Zero-Values</b>	This check box eliminates columns or rows that contain only zeros from the table.
<b>Suppress Missing</b>	This check box eliminates columns or rows that are empty from the table.
<b>Populate Missing Cells</b>	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 activated by default for all pivot tables created in QlikView 7.5 and later.
<b>Null Symbol</b>	The symbol entered here will be used for displaying NULL values in the table.
<b>Missing Symbol</b>	The symbol entered here will be used for displaying missing values in the table.

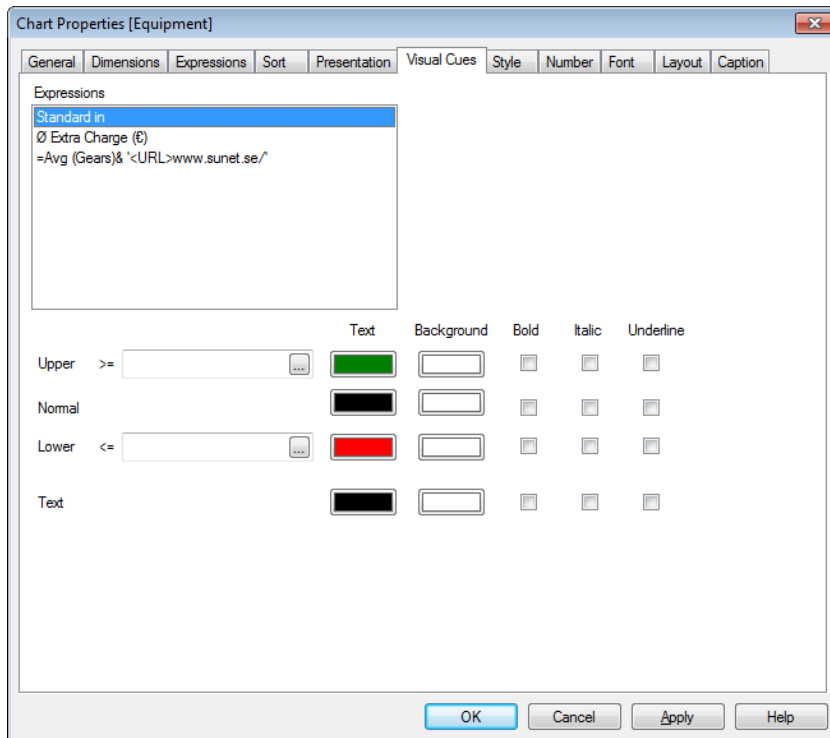
The **Subtotals** group is used for setting the display of totals and subtotals in the pivot table.

<b>Subtotals on Top</b>	If this option is checked, totals will be displayed top/left in the pivot table.
<b>Subtotals at Bottom</b>	If this option is checked, totals will be displayed bottom/right.

In the **Multiline Settings (Expression Data Cells)** group you can specify for values to be displayed in multiple rows, in order to handle longer text strings.

<b>Wrap Header Text</b>	If this option is checked, the contents of a label cell will be displayed in two or more rows. The <b>Header Height _ Lines</b> determines the number of cell lines.
<b>Wrap Cell Text</b>	Same as above, but the setting applies to data cells. The value is set as <b>Cell Height _ Lines</b> .

## 61.8 Chart Properties: Visual Cues



*Chart Properties, Visual Cues*

The **Chart Properties: Visual Cues** page is only available for pivot tables and straight tables. It is opened by a right-click on a chart window and choosing the **Properties** command from the float menu.

Visual cues are used for highlighting expression values and are displayed by applying a different font style, font color and/or cell color. Values belonging to different intervals are typically given different cues.

Values can be specified for three different intervals with the **Upper**  $\geq$ , **Normal** and **Lower**  $\leq$  options, each with different settings. The upper interval specifies values above the numeric value entered in the edit box, the lower interval specifies values below the value entered. The normal values are the values between these two limits. **Text** values are values lacking a valid numerical interpretation.



## 61.9 Chart Properties: Style

Current Style  
Light

Stripes every 0 Rows

Cell Background Color Transparency  
0 % 100 %

Cell Borders Transparency  
0 % 100 %

Border Above Spacing

Background...

### *Chart Properties, Style*

This style page applies to all QlikView tables; table boxes, pivot tables and straight tables. Here you make settings for the table formatting style.

<b>Current Style</b>	Choose an appropriate table style from the drop-down list. If the value [ <b>Custom</b> ] 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.
<b>Stripes every _ Rows</b>	Here you can specify if and at how long intervals shaded stripes should appear.
<b>Indent Mode</b>	This setting is only valid for pivot tables. With this alternative checked, you can achieve a slightly different table style that is especially useful when you need to accommodate a number of dimension labels within a limited table width. <b>Use Only First Dimension Label</b> This setting is only available for pivot tables already in <b>Indent Mode</b> and modifies the style of the pivot table further.
<b>Vertical Dimension Cell Borders</b>	This setting determines whether vertical cell borders are displayed for dimension columns.
<b>Vertical Expression Cell Borders</b>	As above, but for expression columns.
<b>Border Above Spacing</b>	Provided that a <b>Spacing</b> has been determined in the <i>Advanced Field Settings</i> (page 515) dialog, the table style can be slightly modified by checking this alternative.
<b>Background...</b>	Opens the <i>Background Settings</i> (page 489) dialog.
<b>Cell Background Color Transparency</b>	If a color or an image has been applied in <b>Background Settings</b> , you can adjust the transparency of that color or image in the cell background here.

**Cell Borders Transparency** Sets how pronounced the cell borders should be.

## 61.10 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 61.11 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 61.12 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

See *Layout (page 492)*.

## 61.13 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 62 Straight Table

Sales per CategoryName			
CategoryName	ProductName	Sales	Quantity
		<b>\$1,565,525.31</b>	<b>51952</b>
Men's Clothes	Atlas Lussekofta	\$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	\$14,900.64	828
Men's Clothes	Mir2 Trousers	\$17,944.48	1067
Men's Clothes	O-Man Underwear	\$1,649.87	296
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	<b>\$361,096.85</b>	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 Palsii	\$10,472.71	184
Women's Clothes	Okkaba Skin Jackets	\$42,258.78	601
Women's Clothes	Oyaki Kimono	\$9,084.42	806

*An example of a Straight table*

Charts are graphical representations of numerical data. The pivot table and the straight table are special cases as they display the data in table form while retaining all the properties of a chart. It is possible to switch between different representations of an existing chart, by changing the **Chart Type** in the *Chart Properties: General* (page 613) page.

In opposition 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).

The quickest way to create a new straight table is to select *Quick Chart Wizard* (page 767) from the **Tools** menu.

By a right-click on the straight table the *Straight Table: Object Menu* (page 747) will be displayed. It can also be accessed from the **Object** menu, when the straight table is the active object.

For information on the chart types available in QlikView, see *Chart Types* (page 615).

### 62.1 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

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. It is possible to mix dimension and expression columns in any order.

### 62.2 Straight Table: Object Menu

Right-click on a chart and a float menu appears. This menu can also be found under **Object** menu when the chart is active.




The menu contains the following commands:

<b>Properties...</b>	Opens the <b>Properties</b> dialog where the parameters defining the straight table can be set.
<b>Notes</b>	Allows creating and sharing notes about the current object. See <i>Notes and Comments (page 472)</i> for further information.
<b>Detach</b>	<p>The chart title is appended with the text "(Detached)" and the table is no longer updated with selections made in the document (though selections may actually still be made from the chart). The command is available only if the table is attached.</p> <p>By making a copy of a straight table and detaching it, you may make direct comparisons between the copy and the original.</p>
<b>Attach</b>	Attaches a detached straight table. The straight table becomes dynamically linked to the data. The command is available only if the straight table is detached.
<b>Clone</b>	Makes an identical copy of the straight table. If a detached straight table is cloned the clone will be attached.
<b>Fit Columns to Data</b>	Adjusts the width of each table column to its longest data string. Headers are included in the calculations.
<b>Equal Column Width</b>	<p>If the mouse pointer is placed over a column in the straight table this command becomes available in the float menu (not in the main menu bar <b>Object</b> menu).</p> <p>The command sets column width for all columns of the table to that of the column pointed at.</p> <p>Column width can be individually adjusted by moving the pointer to the right edge of the column (pointer changes appearance) and dragging.</p>
<b>Sort</b>	If the mouse pointer is placed over a column in the straight table this command becomes available in the float menu (not in the main menu bar <b>Object</b> menu). The rows of the table will be sorted by the specified column. The sort order is set in the <b>Chart Properties: Sort</b> page.
<b>Custom Format Cell</b>	Opens the <i>Custom Format Cell Dialog (page 524)</i> which lets you format cells in the column and stripe which you clicked upon. This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated.
<b>Order</b>	<p>This cascade menu is only available when the <b>Design Grid</b> command of the <i>View Menu (page 52)</i> is activated or when the <i>Always Show Design Menu Items (page 78)</i> checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127.</p> <p><b>Bring to Front</b> Sets the layout layer of the sheet object to the largest value currently used by any sheet object on the current sheet.</p> <p><b>Send to Back</b> Sets the layout layer of the sheet object to the smallest value currently used by any sheet object on the current sheet.</p> <p><b>Bring Forward</b> Increases the layout layer of the sheet object by one. Maximum value is 127.</p> <p><b>Send Backward</b> Decreases the layout layer of the sheet object by one. Minimum value is -128.</p>

---

<b>Change Value</b>	Only available for expression columns containing an inputsum aggregation of an input field. Sets the cell clicked on in input edit mode. Equivalent to clicking the input icon in the cell.
<b>Restore Values</b>	Only available for expression columns containing an inputsum aggregation of an input field. Opens a cascade menu with three options. <b>Restore Single Value</b> Restores the field values underlying the cell clicked upon on to their default values from the script. <b>Restore Possible Values</b> Restores the values of all possible underlying field values to their default values from the script. <b>Restore All Values</b> Restores the values of all underlying field values to their default values from the script.
<b>Clear All Selections</b>	Clears all selections in the dimensions and expressions of the table.
<b>Print...</b>	Opens the <i>Print: General (page 91)</i> dialog from which you can print the straight table.
<b>Print as PDF...</b>	Opens the <b>Print</b> dialog with the QlikViewPDF printer preselected. After clicking the <b>Print</b> button you will be prompted for a file name for the PDF output file. The QlikViewPDF printer must be downloaded from QlikView's homepage and installed separately.
<b>Send to Excel</b>	Exports the table to Microsoft Excel, which is automatically launched if not already running. QlikView exports in BIFF (native Excel format) format. 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.
<b>Export...</b>	Opens the <b>Save as</b> dialog where path, file name and file type for the exported table content can be specified. The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files).
	<hr/> <b>Note!</b> Mini charts will not be displayed when exporting to Excel! <hr/>
<b>Copy to Clipboard</b>	This cascade menu contains the various copy options for the sheet object. <b>Full Table</b> Copies the table to the clipboard, complete with header and selection status. <b>Table Data Area</b> Copies only the values of the table to the clipboard. <b>Cell Value</b> Copies the text value of the cell right-clicked upon (when invoking the Object menu) to the clipboard. <b>Image</b> 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 <b>User Preferences: Export</b> page. <b>Object</b> 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.

---

<b>Linked Objects</b>	Opens a menu with the following commands for linked objects. <b>Adjust Position of Linked Objects</b> All linked objects on all sheets are adjusted to the same position and size as the one/ones highlighted. <b>Unlink This Object/Unlink Objects</b> This destroys the link between the objects, making them different objects with different object IDs.
<b>Minimize</b>	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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Maximize</b>	Enlarges the object 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 object's <b>Properties</b> dialog on the <i>Caption (page 496)</i> page.
<b>Restore</b>	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.
<b>Help</b>	Opens QlikView help.
<b>Remove</b>	Removes the active sheet object from the sheet.

## 62.3 General

On the **General** page you set the chart type, choose a name for the chart etcetera. This page is the same for all chart types. See *Chart Properties: General (page 613)*.

## 62.4 Dimensions

On the **Dimensions** page you set the dimensions to be shown in the chart. This page is identical to *Chart Properties: Dimensions (page 620)*.

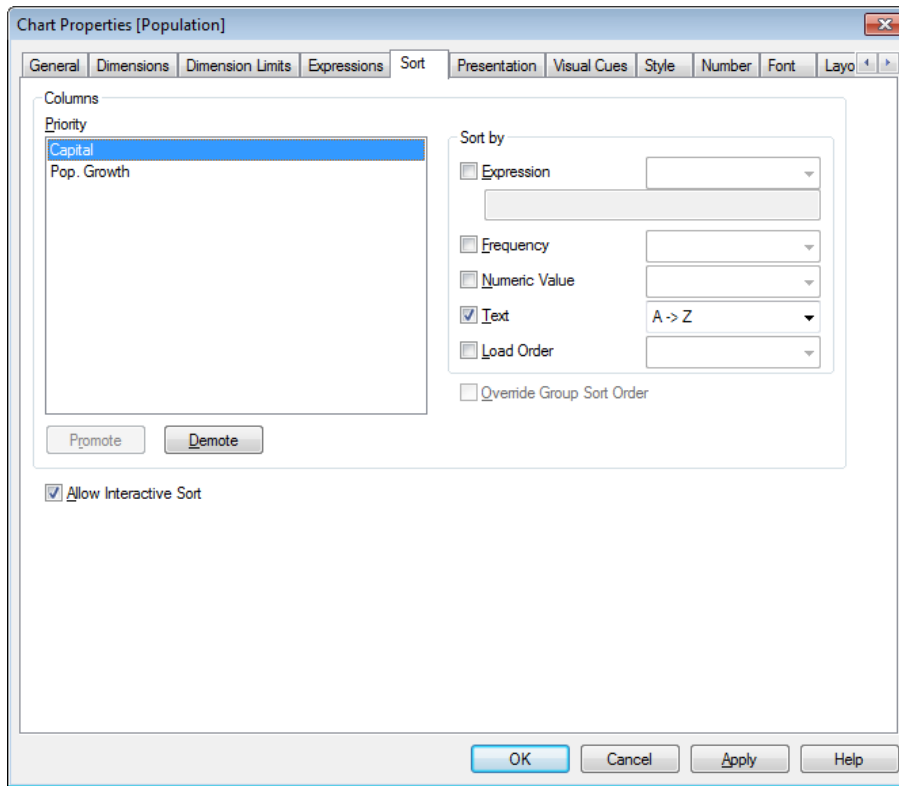
## 62.5 Dimension Limits

On the **Dimension Limits** page controls the number of dimension values you can see in a given chart. This page is identical to *Chart Properties: Dimension Limits (page 625)*.

## 62.6 Expressions

On the **Expressions** page you set the expressions to be displayed in the chart. This page is identical to *Chart Properties: Expressions (page 629)*.

## 62.7 Chart Properties: Sort (Straight Table)



*Chart Properties, Sort (Straight Table)*

The **Chart Properties: Sort** page is reached by a right-click on a straight table and selecting **Properties** from the **Object** menu.

This is where you decide the sort order of the chart dimension(s) from a number of available sort orders.

The variables and expressions that define the straight table are listed in the **Columns** group.

**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. It is also possible to change the sort priority by clicking the **Promote** and **Demote** buttons.

**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.

**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**

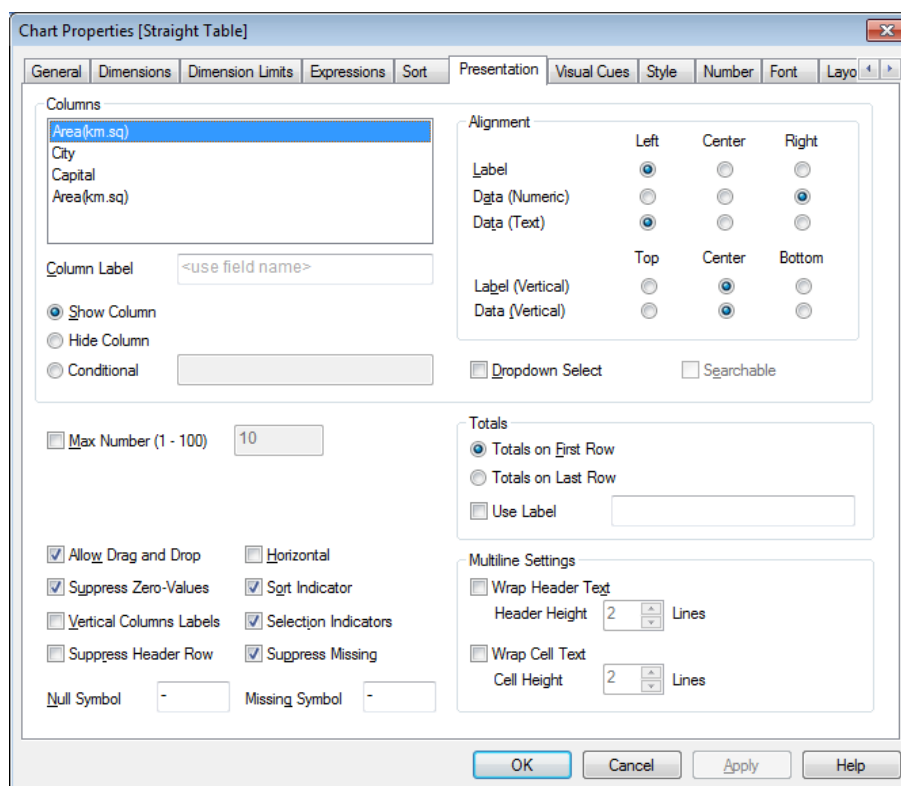
Sorts the column values according to their initial load order.

**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.

**Allow Interactive Sort** Disable the **Object - Sort** command by unchecking this option.

## 62.8 Chart Properties: Presentation (Straight Table)



*Chart Properties, Presentation (Straight Table)*

The dimensions and expressions that define the straight table are listed in the **Columns** group. Selecting one column makes it possible to change its settings.

**Column Label** The column label displays the label of the field, which may also be edited.

**Show Column** By selecting this radio button, the selected column will be visible when the table is drawn in the layout.

**Hide Column** In some situations you may want to sort a straight table on a field that is not to be visible to the user. The **Hide Column** option lets you hide a field while still having it available for sorting purposes. The sort criteria are stated in the **Priority** of the **Column** group on *Chart Properties: Sort (Straight Table)* (page 751)page.



---

<b>Conditional</b>	The column will be shown or hidden depending on a condition 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 <i>Edit Expression Dialog (page 757)</i> is opened for easier editing of long formulas
<b>Alignment</b>	In this group, the alignment of the dimension values, expression values and their labels within the straight table can be set. <b>Label</b> , <b>Data (Numeric)</b> and <b>Data (Text)</b> can be individually set to <b>Left</b> , <b>Center</b> or <b>Right</b> . When multi line cells and labels are used, <b>Label (Vertical)</b> and <b>Data (Vertical)</b> can be set to <b>Top</b> , <b>Center</b> or <b>Bottom</b> .
<b>Dropdown Select</b>	The <b>Dropdown Select</b> option adds a drop-down arrow icon to the left side of the header of any selected column. Click the icon to access the data from a drop-down data list. This is very similar to making selections in a multi box.
<b>Searchable</b>	If the checkbox <b>Searchable</b> is selected for an expression column, a search icon will appear to the left in the column header. Click the icon and a search box opens. Then type a search criterion (e.g. >100 000). When you hit Enter, all table rows with an expression value matching the search criteria will be selected.
<b>Max Number (1 - 100)</b>	Denotes the maximum number of rows to be displayed.
<b>Allow Drag and Drop</b>	With this alternative it will be possible to sort the sequence of the fields of the table by clicking and dragging the headers. It is possible to mix dimension and expression columns in any order.
<b>Horizontal</b>	With this option checked, the straight table is transposed 90 degrees, so that the data columns are displayed horizontally.
<b>Suppress Zero-Values</b>	The check box <b>Suppress Zero-Values</b> eliminates expressions that contain only zeros or null values from the table.
<b>Sort Indicator</b>	This option adds a sort indicator to the right side of the column header of the field that the table is currently sorted by. The icon is flipped to reflect ascending or descending order.
<b>Vertical Columns Labels</b>	Labels for the column headers will be rotated to vertical.
<b>Selection Indicators</b>	With this option checked, a colored indicator is displayed in the header of any field column where selections have been made.
<b>Suppress Header Row</b>	With this option checked, the table is displayed without header (label) row.
<b>Suppress Missing</b>	This option is not functional on straight tables.
<b>Null Symbol</b>	The symbol entered here will be used for displaying NULL values in the table.
<b>Missing Symbol</b>	The symbol entered here will be used for displaying missing values in the table.

---

**Totals**

Totals are shown for the expression columns, according to the settings below.

**Totals on First Row**

Totals are displayed near the top of the table.

**Totals on Last Row**

Totals are displayed at the end of the table.

**Use Label**

Here you may enter a label for totals.

**Multiline Settings**

In this group the table header and data cells may be set to display values in multiple rows, which is useful for long text strings.

**Wrap Header Text**

With this option selected, the header will display its contents in more than one row.

**Header Height \_ Lines**

Here you specify a limit to the number of header lines.

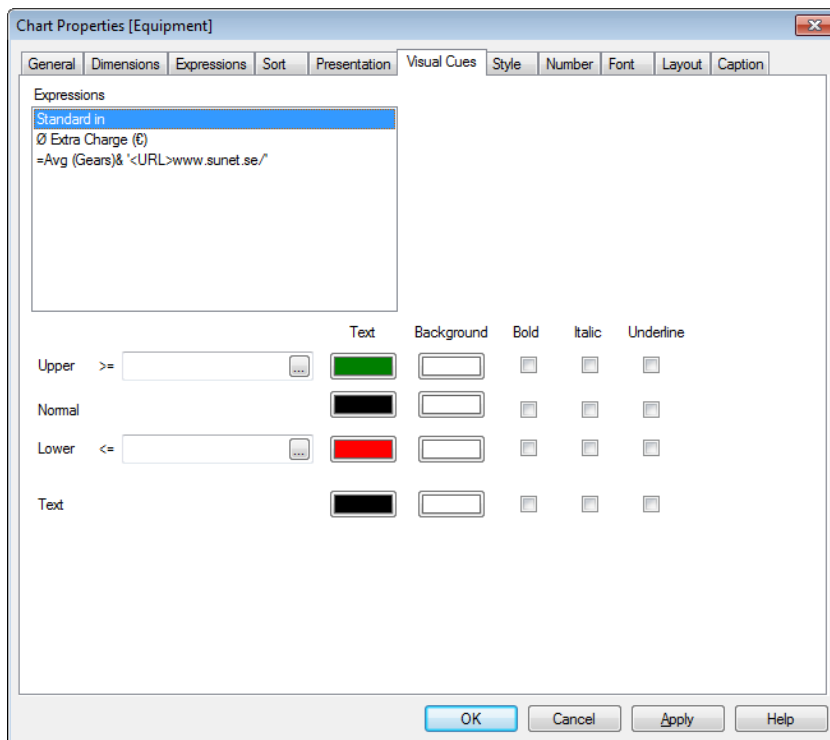
**Wrap Cell Text**

With this option selected, a cell will display its contents in more than one row.

**Cell Height \_ Lines**

Here you specify a limit to the number of cell rows.

## 62.9 Chart Properties: Visual Cues



*Chart Properties, Visual Cues*

The **Chart Properties: Visual Cues** page is only available for pivot tables and straight tables. It is opened by a right-click on a chart window and choosing the **Properties** command from the float menu.

Visual cues are used for highlighting expression values and are displayed by applying a different font style, font color and/or cell color. Values belonging to different intervals are typically given different cues.

Values can be specified for three different intervals with the **Upper**  $\geq$ , **Normal** and **Lower**  $\leq$  options, each with different settings. The upper interval specifies values above the numeric value entered in the edit box, the lower interval specifies values below the value entered. The normal values are the values between these two limits. **Text** values are values lacking a valid numerical interpretation.

## 62.10 Chart Properties: Style

### *Chart Properties, Style*

This style page applies to all QlikView tables; table boxes, pivot tables and straight tables. Here you make settings for the table formatting style.

- |   |   |
|---|---|
| <b>Current Style</b>                    | Choose an appropriate table style from the drop-down list. If the value [ <b>Custom</b> ] 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.   |
| <b>Stripes every _ Rows</b>             | Here you can specify if and at how long intervals shaded stripes should appear.   |
| <b>Indent Mode</b>                      | This setting is only valid for pivot tables. With this alternative checked, you can achieve a slightly different table style that is especially useful when you need to accommodate a number of dimension labels within a limited table width.<br><b>Use Only First Dimension Label</b><br>This setting is only available for pivot tables already in <b>Indent Mode</b> and modifies the style of the pivot table further. |
| <b>Vertical Dimension Cell Borders</b>  | This setting determines whether vertical cell borders are displayed for dimension columns.  |
| <b>Vertical Expression Cell Borders</b> | As above, but for expression columns.   |
| <b>Border Above Spacing</b>             | Provided that a <b>Spacing</b> has been determined in the <i>Advanced Field Settings</i> (page 515) dialog, the table style can be slightly modified by checking this alternative.  |

<b>Background...</b>	Opens the <i>Background Settings (page 489)</i> dialog.
<b>Cell Background Color Transparency</b>	If a color or an image has been applied in <b>Background Settings</b> , you can adjust the transparency of that color or image in the cell background here.
<b>Cell Borders Transparency</b>	Sets how pronounced the cell borders should be.

## 62.11 Number

On the **Number** page you can set the display format for numbers. This page is similar to the **Number** page of the bar chart.

See *Chart Properties: Number (page 655)*.

## 62.12 Font

On the **Font** tab you can change the font of the text in the object. This page is identical to the **Font** page of the list box.

See *Font (page 491)*.

## 62.13 Layout

On the **Layout** tab you can specify how the sheet object should appear on the layout. This includes settings for shape, border and the layer on which the object should reside. This page is identical to the **Layout** page of the list box.

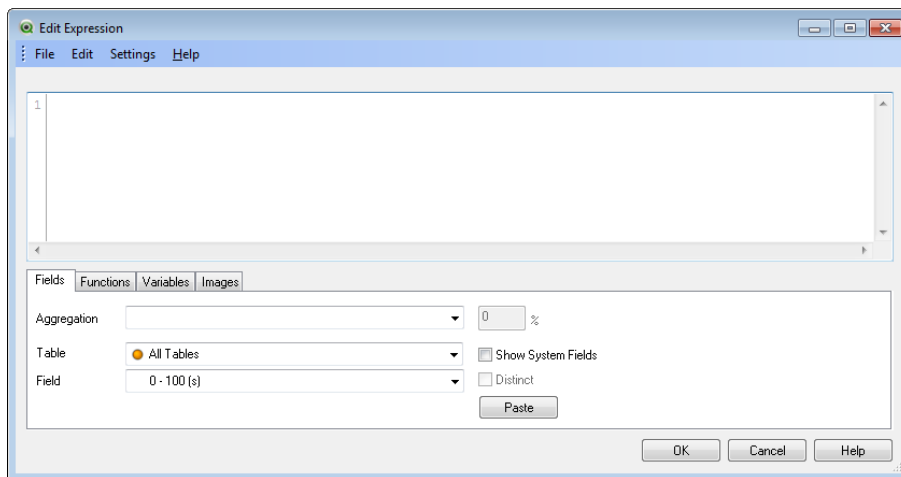
See *Layout (page 492)*.

## 62.14 Caption

On the **Caption** tab you can make advanced settings for the caption, including background and foreground (text) color with separate settings for active and inactive state. This page is identical to the **Caption** page at object, document and sheet level.

See *Caption (page 496)*.

## 63 Edit Expression Dialog



*The Edit Expression dialog*

This dialog is opened by selecting the **Add** button below the expressions list in *Chart Properties: Expressions* (page 629) or by right-clicking an existing expression and selecting **Add Expression** or **Edit Expression**.

Though it is primarily used in chart expressions, the same dialog is also utilized for building or editing other expressions wherever *Calculated Formula* (page 871) are needed. It is universally accessed by clicking the button with the three dots symbol that can be found next to text edit boxes throughout the program.

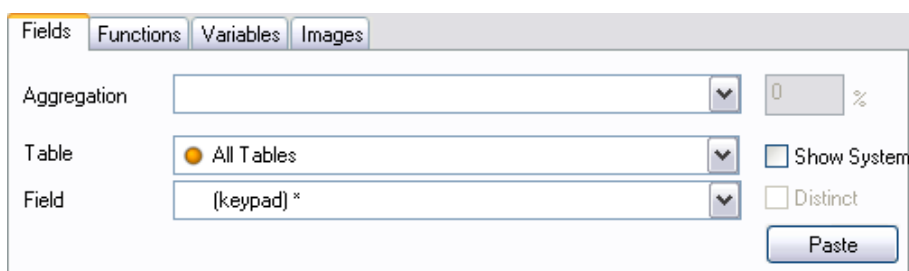
The main parts of the dialog are made up of the **Expression** edit area that can be resized to accommodate large expressions and beneath this, the tab area consisting of the four tabs **Fields**, **Functions**, **Variables** and **Images**.

### Expression OK

This is the edit area for the selected expression. Type in the whole expression here, but it is often convenient to do most of the expression building in the tab area described below.

The label **Expression OK** is only displayed as long as the expression syntax is accepted by the program, i.e. while the expression is valid. Otherwise the label will display one of several error messages (**Bad fieldname(s)**, **Error in expression**, **Garbage after expression**). There is little point in leaving the dialog by clicking the **OK** button, unless the expression is valid.

### Fields



The **Fields** tab holds controls that are used for creating statistical aggregation functions based on field data.

<b>Aggregation</b>	The first drop down contains a list of available statistical aggregation functions. (The functions <b>First String</b> and <b>Last String</b> can e.g. be used for showing text in a pivot table.)
<b>Table</b>	This drop-down list allows you to limit the fields displayed in the <b>Field</b> list below to one specific internal table.
<b>Field</b>	This drop down contains a list of available field names. Fields loaded as MEASURE using the <i>Direct Discovery (page 161)</i> function are marked with <i>[Measure]</i> . MEASURE fields can only be used with <i>Aggregation Functions (page 298)</i> Sum, Avg, Count, Min, and Max. For more advanced use or database specific functions you can use the <b>SQL</b> statement to execute a valid SQL function, with the limitation that you can only reference a table or view loaded with Direct Discovery .
<b>Show System Fields</b>	Check this alternative in order to include the system fields in the <b>Field</b> list.
<b>Distinct</b>	Statistical functions are by default calculated on number of occurrences. By checking the <b>Distinct</b> option, the expression will be calculated without duplicate values. (For a better explanation of the use of the <b>distinct</b> qualifier, see the <i>Chart Aggregation functions (page 790)</i> ).
<b>Paste</b>	Click this button in order to enter your selections as a component of the expression.

## Functions



The **Functions** tab holds controls that are used for entering general QlikView-functions into the expression.

<b>Function Category</b>	This drop-down list allows you to limit the functions displayed in the <b>Function Name</b> list below to a single category at a time.
<b>Function Name</b>	This drop down contains a list of all available QlikView-functions (except a limited number of dedicated script functions). The pane at the bottom of the functions tab displays the argument syntax of the selected function.
<b>Paste</b>	Click this button in order to enter your selections as a component of the expression.

## Variables

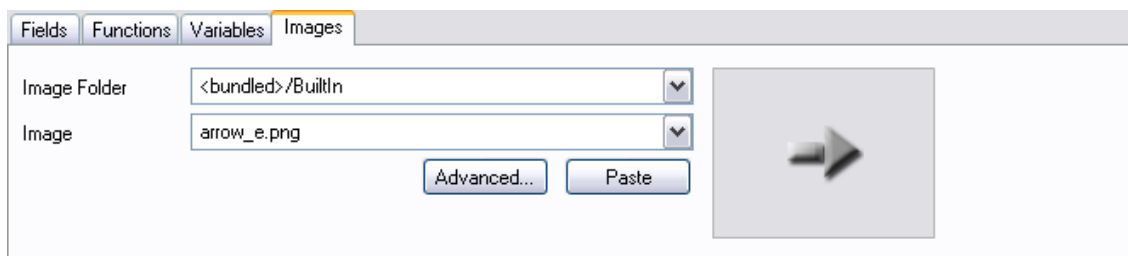


The **Variables** tab holds controls that are used for entering QlikView-variables into the expression.

- Variables** This drop down contains a list of all variables currently defined for the document.
- Show System Variables** Check this alternative in order to include the system variables in the **Variables** list.
- Paste** Click this button in order to enter your selections as a component of the expression.

At the bottom of the **Variables** tab is a pane displaying the current value of any variable selected in the **Variables** list.

## Images



The **Images** tab holds controls that are used for accessing the *Internal Files (page 157)* that are available in QlikView. It is also possible to access other associated image files. Note that this functionality is limited to certain parts of the layout.

- Image Folder** This drop down contains a list of available image folders.
- Image** This drop down contains a list of available images in the selected folder.
- Advanced...** Click this button to select directly from the pictures in the **Image Selector** dialog.
- Paste** Click this button in order to enter your selections as a component of the expression.

To the right in the **Images** tab is a pane displaying the currently selected image.

In addition, the **Edit Expression** dialog contains the following menu commands and buttons.

### 63.1 File Menu

- Export to Expression File...** It is possible to save the contents of the **Expression** edit box as a table file by means of the **Export to Expressions File** command. The file is saved with the extension **.qve**.
- Insert File...** A previously created expression file can be inserted into the script by means of the **Insert File...** command.
- Colormix Wizard...** A color mix expression can be created using the *Colormix Wizard (page 760)*.
- Print** Opens Windows standard **Print** dialog for printing the expressions. This command can also be invoked by the following keyboard shortcut: Ctrl+P.

## 63.2 Edit Menu

<b>Undo</b>	Undoes the latest change. This command can also be invoked by the following keyboard shortcut: Ctrl+Z.
<b>Redo</b>	Redoes the latest <b>Undo</b> . This command can also be invoked by the following keyboard shortcut: Ctrl+Y.
<b>Cut</b>	Exports the highlighted text to the clipboard. This command can also be invoked by the following keyboard shortcut: Ctrl+X.
<b>Copy</b>	Copies the highlighted text to the clipboard. This command can also be invoked by the following keyboard shortcut: Ctrl+C.
<b>Paste</b>	Pastes the contents of the clipboard into the dialog at the position of the cursor. This command can also be invoked by the following keyboard shortcut: Ctrl+V.
<b>Clear</b>	Clears the entire expression.
<b>Select All</b>	Selects the entire expression.
<b>Find/Replace...</b>	Opens a dialog where it's possible to find and replace numbers or characters in the expression. This command can also be invoked by the following keyboard shortcut: Ctrl+F.

## 63.3 Settings Menu

<b>Configure...</b>	Opens <i>User Preferences: Editor (page 74)</i> .
---------------------	---

## 63.4 Colormix Wizard

With the **Colormix Wizard**, opened from the **File** menu in the **Edit Expression** dialog, it is possible to create a colormix 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.

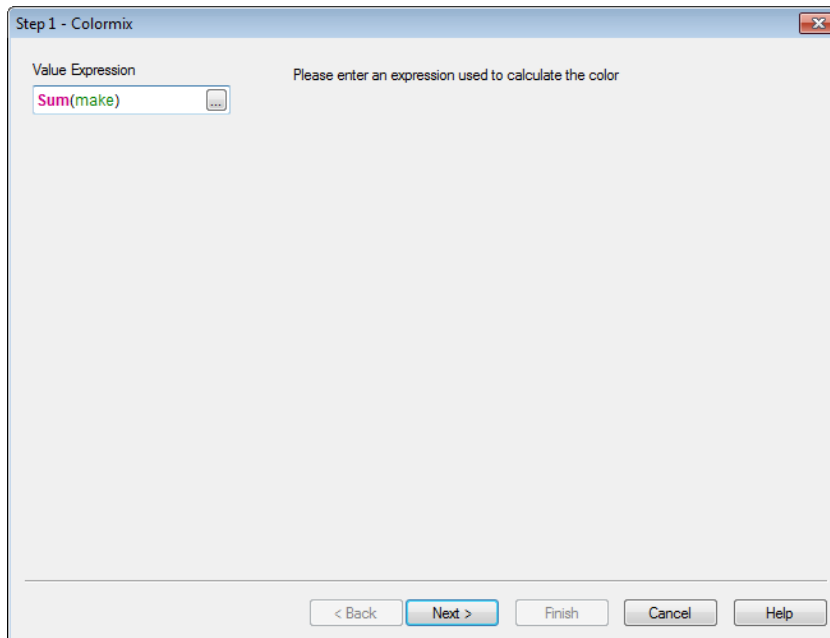
First, it is important to clarify that the color functions, hence also the **Colormix Wizard**, are only relevant to use in place where QlikView expects a color function, i.e. not in the chart expression itself, but rather in the **Background Color** expression or in the **Text Color** expression.

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.

Click **Next** to continue.



## Step 1 – Enter a Value Expression



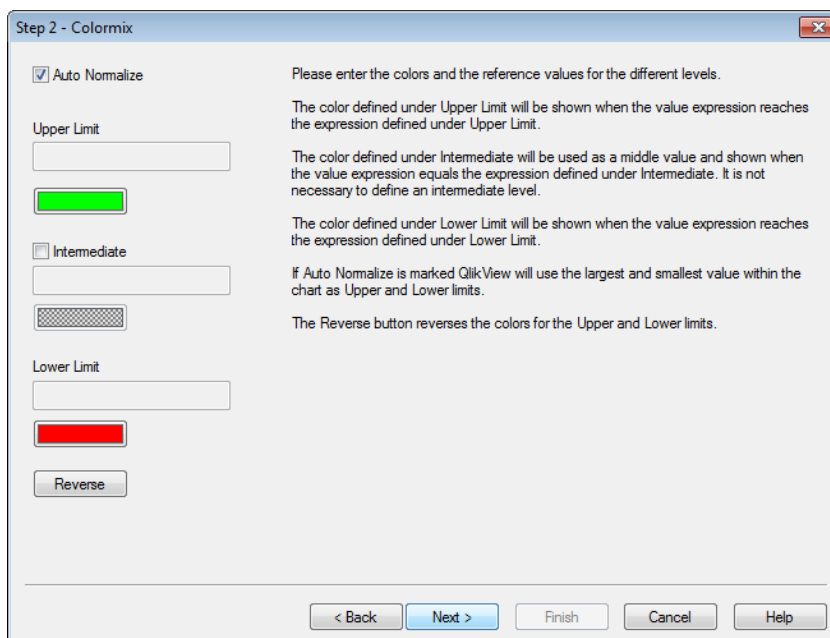
This is the expression that will control what color QlikView will show. Typical expressions could be

- $\text{Sum}(\text{Sales}) / \text{Sum}(\text{total Sales})$
- $\text{Sum}(\text{Sales}) / \text{Sum}(\text{Quota})$
- $\text{Avg}(\text{Age})$

Note that it is usually a value that has the same order of magnitude no matter how many or how few records that you have selected. An average, a percentage or an index is usually a good measure to use.

Click **Next** to continue.

## Step 2 – Set Upper and Lower Limits



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
Sum(Sales) / Sum(total Sales)	Sum(total Sales)	0
Sum(Sales) / Sum(Quota)	2 (=200%)	0
Avg(Age)	Max(total Age)	Min(total Age)

But also other limits are of course possible.

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.

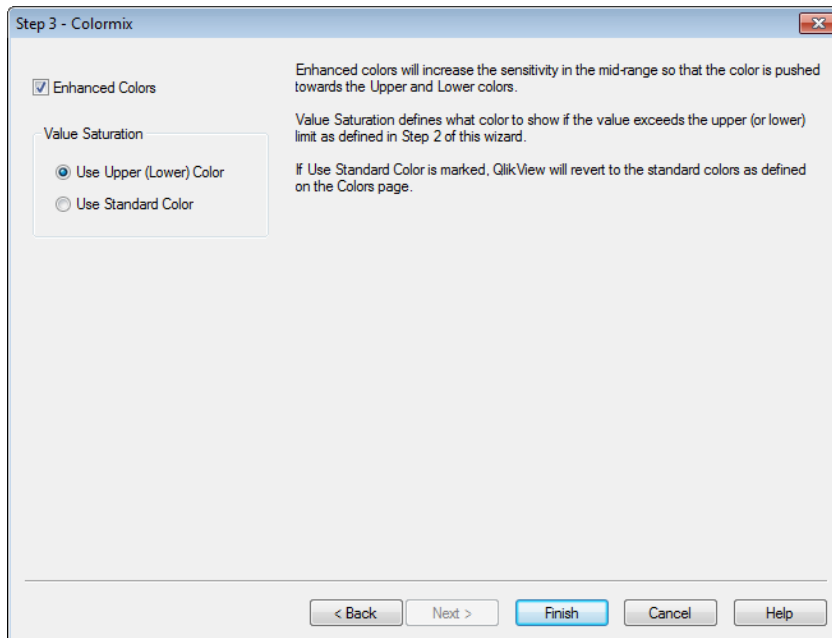
**Intermediate** Here it is possible to enter an expression for an intermediate level, linked to a third color.

**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 3 – Finalize



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.



## 64 Boxplot 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.

### 64.1 Boxplot Wizard Define Data

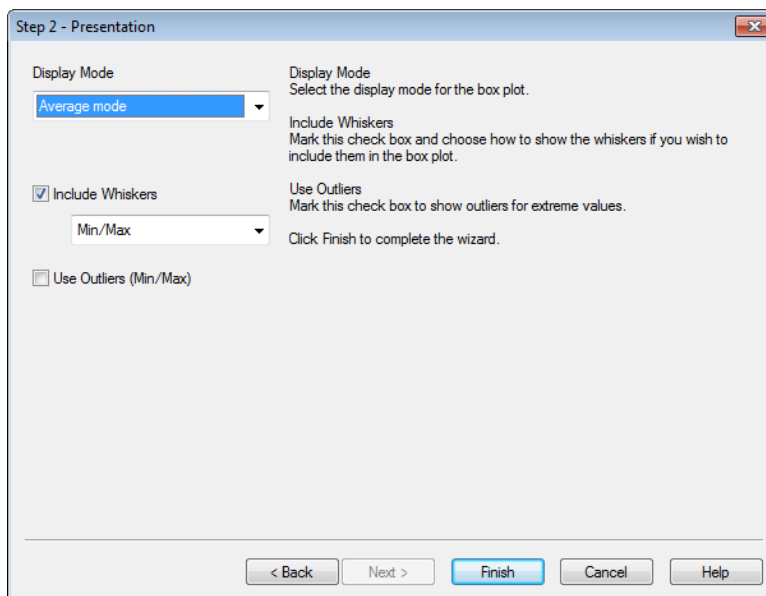
Select the **dimension** for the x-axis. After finishing this wizard you can add more dimensions or change dimensions at any time via the **Chart Properties** dialog.

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. Expressions 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.

## 64.2 Boxplot Wizard Presentation

**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.

## 65 Quick Chart Wizard

The Quick Chart feature has been developed for users who want to create a simple chart in a quick and easy way, without bothering about the great number of different settings and options available.

The Quick Chart Wizard takes you through the following basic steps:

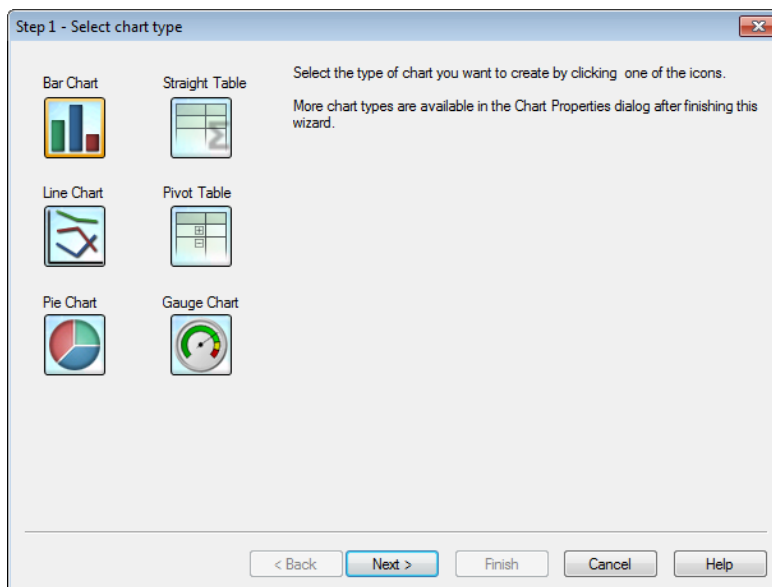
- 1) Select chart type
- 2) Define chart dimension(s)
- 3) Define the chart expression
- 4) Format chart

Depending on selections, some of the above steps may be skipped.

### 65.1 Starting the Quick Chart Wizard

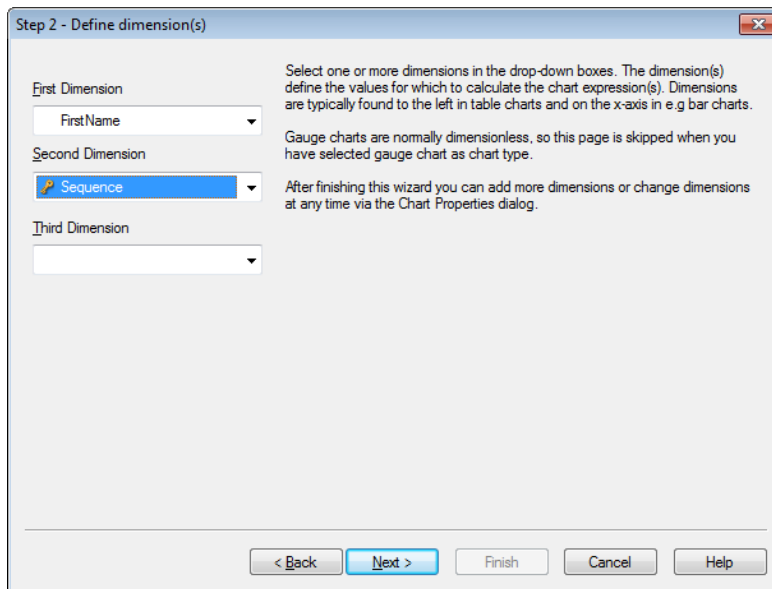
Start the Quick Chart Wizard either by choosing **Quick Chart Wizard** from the **Tools** menu or by clicking on the corresponding button in the **Design** toolbar.

The first time the wizard is started a start page will open outlining the purpose of the wizard and the basic steps involved. If this start page should be skipped when the wizard is used in the future, mark the **Don't show this page again** check box. Click **Next** to continue. The Quick Chart Wizard goes through the following basic steps:



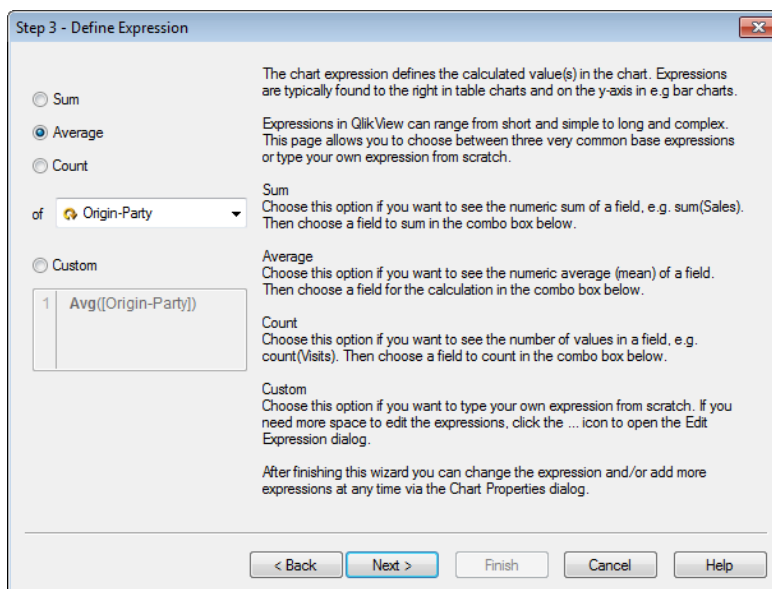
#### Step 1: Select chart type

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. It is possible to 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)

Unless you selected Gauge chart in the first step, you will now see the **Define Dimension** page. This page is automatically skipped for gauge charts because they do not normally have any dimensions. 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 the 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

The chart expression defines the calculated value(s) in the chart. Expressions are typically found to the right in table charts 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

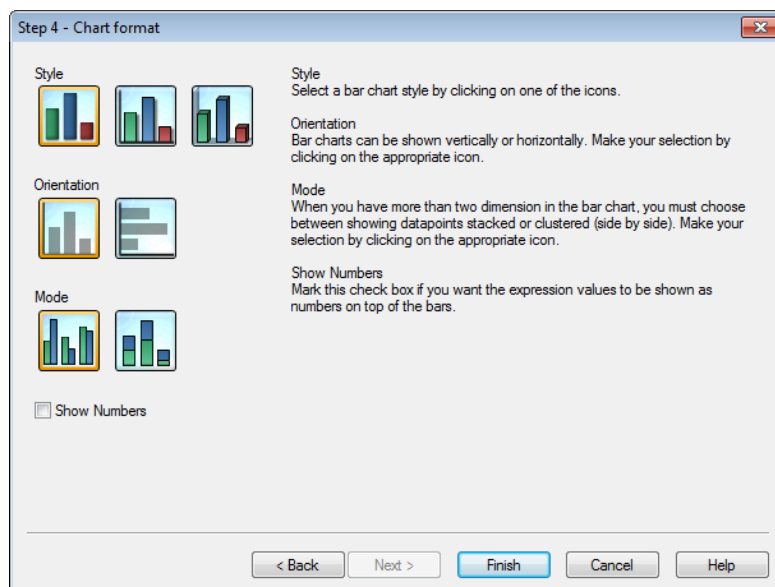
Choose this option if you want to see the number of values in a field, e.g. count(OrderID). Then choose a field to count in the combo box below.

#### 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)

In this page you set formatting options for the selected chart type.

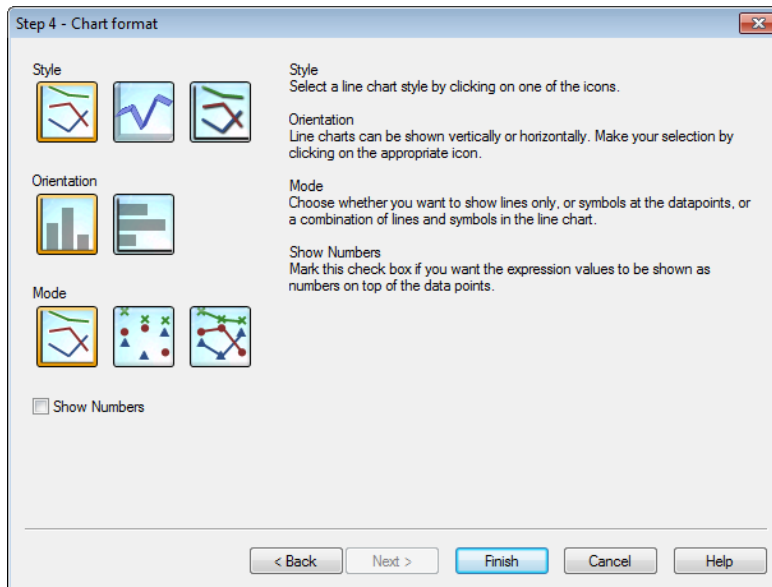
##### Mode

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 on 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 can return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.



#### Step 4: Chart format (line chart)

In this page you set formatting options for the selected chart type.

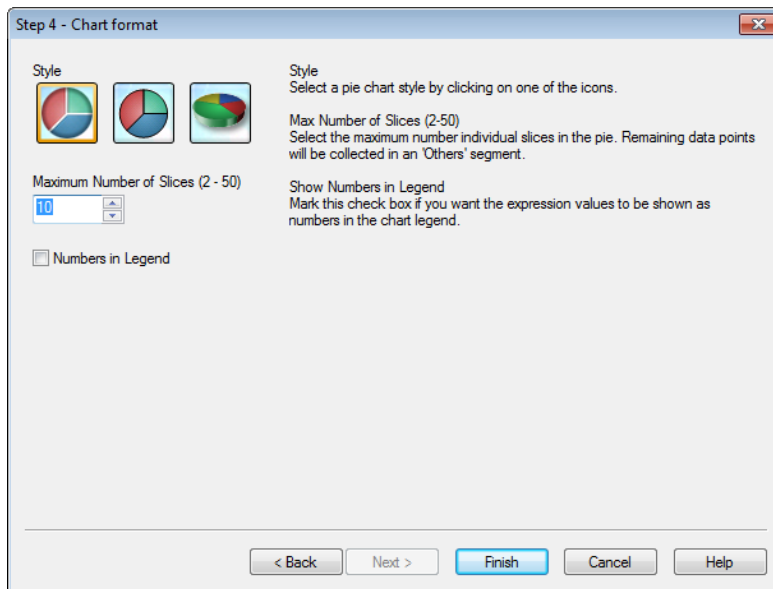
##### Mode

Show data points as lines, symbols or a combination thereof. Make the selection by clicking on the appropriate icon.

##### Show Numbers

Mark this check box if the expression values is 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 it is possible to return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.



#### Step 4: Chart format (pie chart)

In this page you set formatting options for the selected chart type.

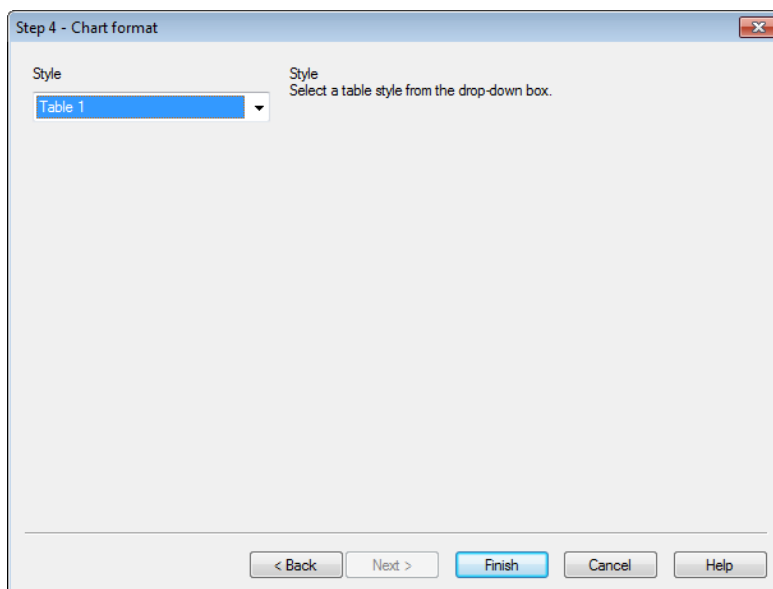
##### **Max Number of Slices (2-50)**

Select the maximum number of individual slices in the pie. Remaining data points will be collected in an '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 can return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.



#### Step 4: Chart format (straight table)

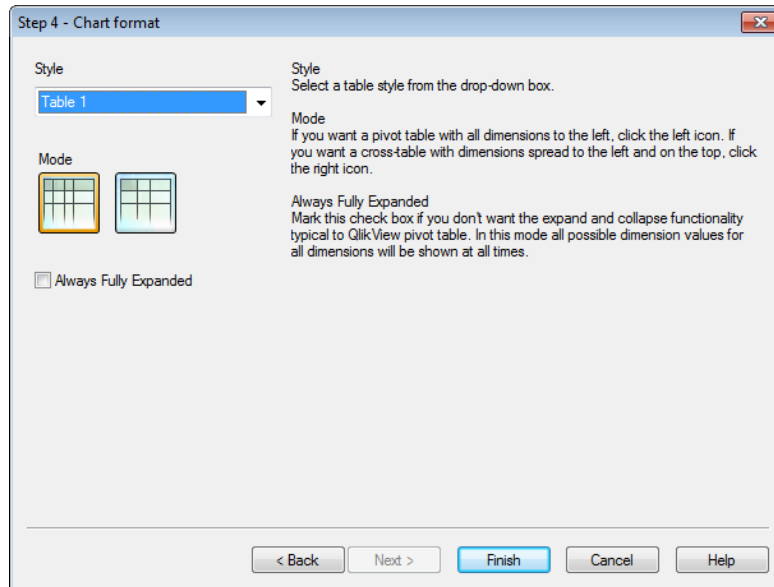
In this page you set formatting options for the selected chart type.

##### 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 can return to the chart at any time via the regular

**Chart Properties** dialog to make further adjustments.



#### Step 4: Chart format (pivot table)

In this page you set formatting options for the selected chart type.

##### Style

Select a table style from the drop-down box.

##### Mode

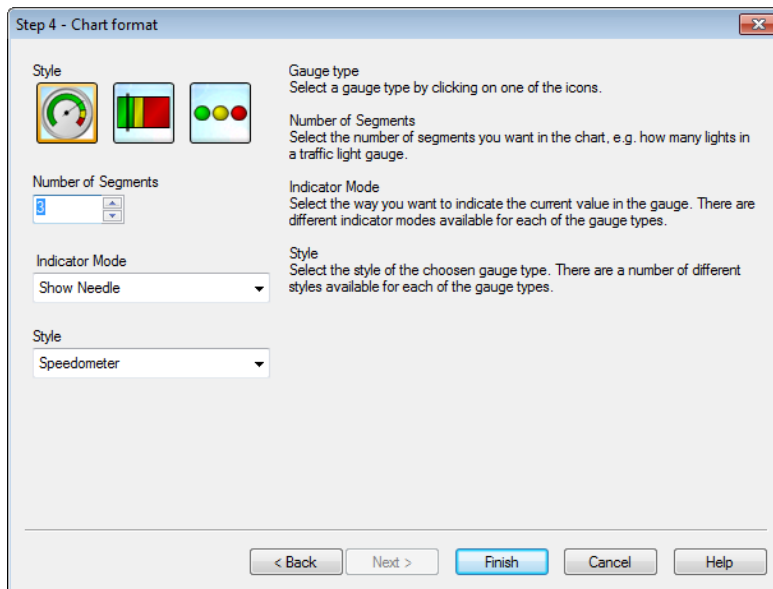
If you want a pivot table with all dimensions to the left, click on the left icon. If you want a cross-table with dimensions spread to the left and on the top, click on the right icon.

##### Always Fully Expanded

Mark this check box if you don't want the expand and collapse functionality typical to a 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 can return to the chart at any time via the regular

**Chart Properties** dialog to make further adjustments.



#### Step 4: Chart format (gauge chart)

In this page you set formatting options for the selected chart type.

##### **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.

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.



## 66 Time Chart Wizard

The time chart wizard helps you with the common task of building 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 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.

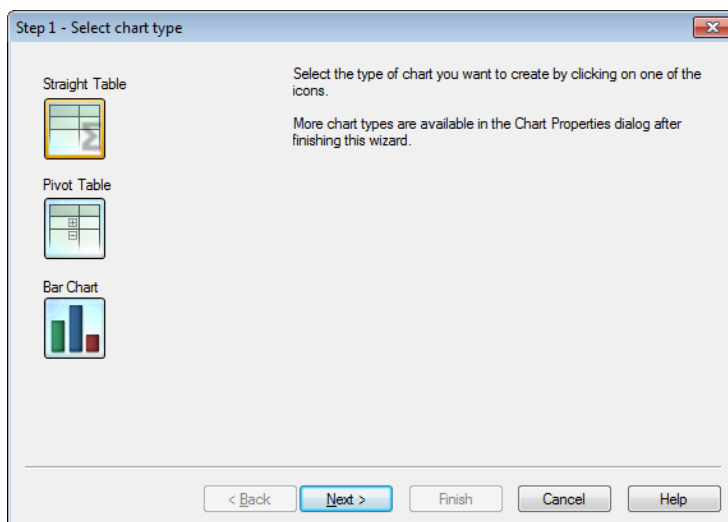
### 66.1 Starting the Time Chart Wizard

Start the time chart wizard either by choosing **Time Chart Wizard** from the **Tools** menu or by clicking on the corresponding button in the **Design** toolbar.

The first time you start the wizard you are 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.

Depending on selections some of the below steps may be skipped. Move forward in the wizard by clicking **Next**. Click **Finish** in order to finalize the chart and return to the layout. Return to the chart at any time via the regular **Chart Properties** dialog in order to make further adjustments.

The time chart wizard takes you through the following basic steps:



#### Select chart type

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. It is possible to change the chart into any other QlikView chart type via the **Chart Properties** dialog after finishing the wizard.

**Step 2 - Define 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.

First Dimension: Author

Second Dimension: [Empty]

Third Dimension: [Empty]

< Back   Next >   Finish   Cancel   Help

### Define dimension(s)

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.

**Step 3 - Expression and 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 the ... icon to open the Edit Expression dialog.

Base Expression: [Empty]

Timestamp Expression: [Empty]

Basedate Expression: Today()

< Back   Next >   Finish   Cancel   Help



**Expression and time references**

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).

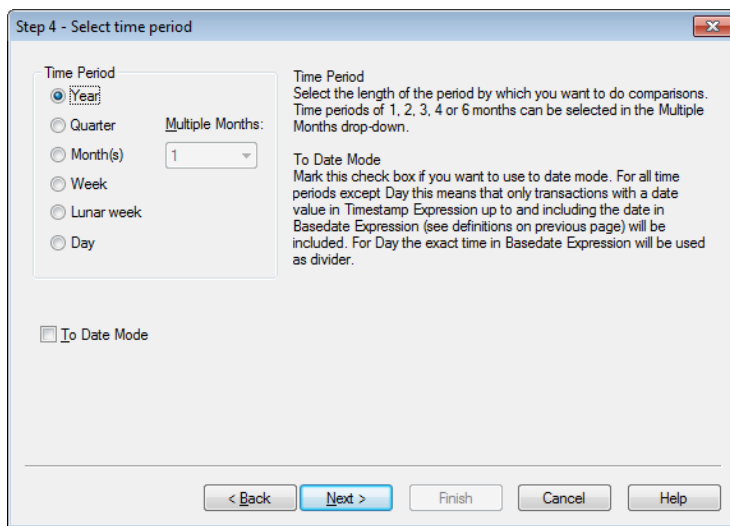
**Timestamp Expression**

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 which 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.

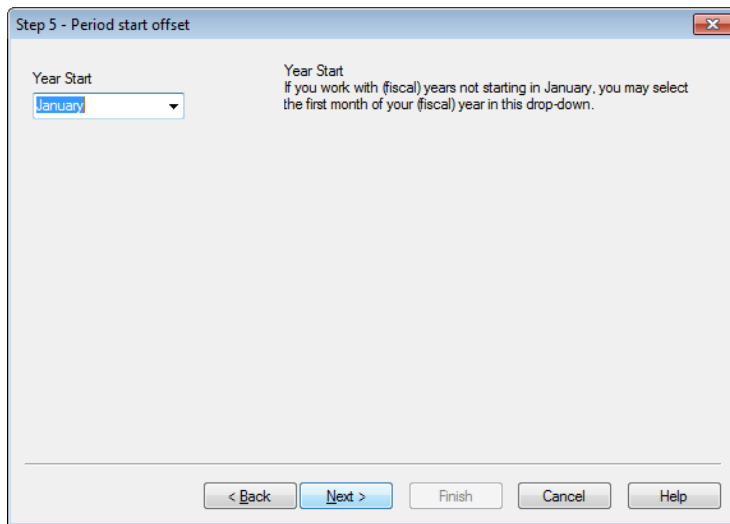
**Select time period**

In this page you select the basic time period unit for which the chart will be made.

Under **Time Period** you select the length of the period by which you want to qualify the chart expression and make 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, meaning that lunar week 52 will contain 8 or 9 days.

Mark the check box **To Date Mode** if you want to use to 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.

If you mark the checkbox **Last Completed Period as Base**, QlikView will use the last completed period as base, e.g. the last complete quarter.



### Period start offset

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:

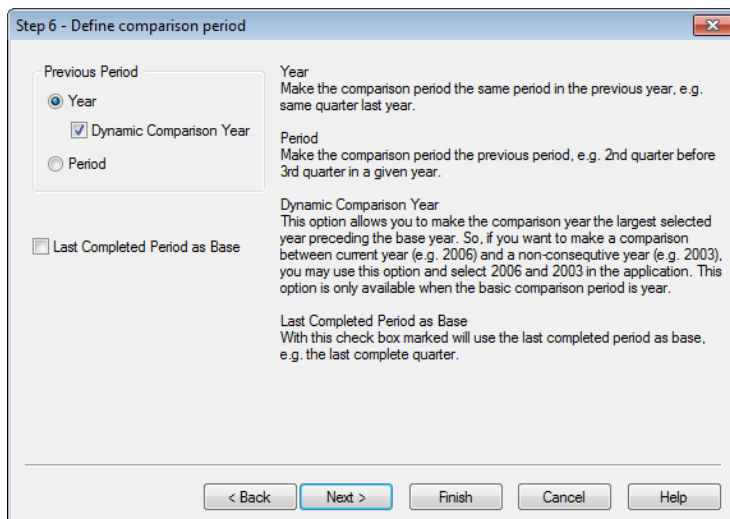
#### Year Start

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** or **Months**.

#### Week/Day offset

If you want to offset the start of the week or the start of the day you may do so 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**.



### Define comparison period

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

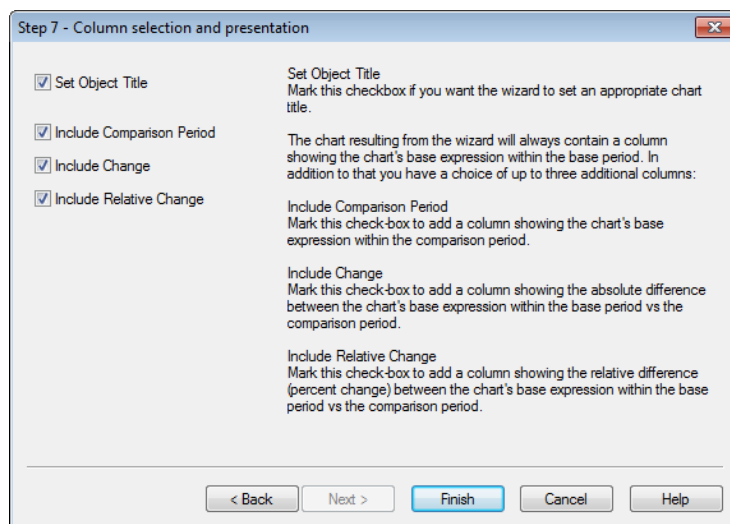
Make the comparison period the same period in the previous year, e.g. same quarter last year.

#### 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 current year (e.g. 2006) and a non-consecutive year (e.g. 2003), you may use this option and the select 2006 and 2003 in the application. This option is only available when the basic comparison period is year.



### Column selection and presentation

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.

Mark the checkbox **Set Object Title** 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 in order to add a column showing the chart's base expression within the comparison period.

#### Include Change

Mark this check-box in order to add a column showing the absolute difference between the chart's base expression within the base period vs the comparison period.

#### Include Relative Change

Mark this check-box in order to add a column showing the relative difference (percent change) between the chart's base expression within the base period vs. the comparison period.



## 67 Statistics Chart Wizard

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

### 67.1 Starting the Statistics Chart Wizard

Start the statistics chart wizard by choosing **Statistics Chart Wizard** from the **Layout** menu. The first page offers 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. Read more about this under *Statistical Test Functions in Charts (page 805)* and *Statistical Test Functions in Script (page 307)*. For further information about statistical tests and their uses, please refer to a statistics text book.

### 67.2 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. To obtain the Chi2-test chart, you must specify the following:

<b>Column</b>	Specify a field or expression returning column headers for the test data sample.
<b>Row</b>	Specify a field or expression returning row headers for the test data sample.
<b>Value</b>	Specify a field or expression returning test value data.
<b>Expected Value</b>	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.
<b>p (Show p-value / Significance)</b>	Mark this check box to include the <b>p</b> 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. Return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

## 67.3 Paired T-Test

This option generates a straight table with values returned from a paired student's t-test.

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. Return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

## 67.4 Independent Samples T-Test

This option generates a straight table with values returned from a two-sample unpaired student's t-test.

To obtain the t-test chart, you must specify the following:

- |   |   |
|---|---|
| <b>Grouping Field/Expression</b>                  | Specify a field or expression returning test grouping data series. Exactly two values must be returned.   |
| <b>Test Field/Expression</b>                      | Specify a field or expression returning the test value data series.   |
| <b>t (Show t Value)</b>                           | Mark this check box to include the t value in the resulting chart.  |
| <b>df (Show Degrees of Freedom)</b>               | Mark this check box to include the df value (degrees of freedom) in the resulting chart.  |
| <b>p (Show p-value / Significance (2-tailed))</b> | Mark this check box to include the p-value (significance, 2-tailed) in the resulting chart.   |
| <b>Mean difference</b>                            | Mark this check box to include the mean difference value in the resulting chart.  |
| <b>Lower (CL %)</b>                               | Mark this check box to include the lower limit of the confidence interval (as specified in <b>Confidence Level</b> below) in the resulting chart. |
| <b>Upper (CL %)</b>                               | Mark this check box to include the upper limit of the confidence interval (as specified in <b>Confidence Level</b> below) in the resulting chart. |
| <b>Confidence Level</b>                           | Specify a percentage for the level of confidence.   |

Click **Finish** to finalize the chart and return to the layout. Return to the chart at any time via the regular **Chart Properties** dialog to make further adjustments.

## 67 How To Create Charts with Direct Discovery

Charts can be created with DIMENSION fields as chart dimensions and MEASURE fields in chart expressions. DETAIL fields have no function in charts and cannot be used.

For QlikView charts that use only Direct Discovery fields, all of the aggregations are performed in the database. When a chart has both Direct Discovery and in-memory fields, the database aggregations are performed first and then the chart-level aggregation are performed after the database aggregations complete.

The following *Aggregation Functions* (page 298) can be used with MEASURE fields:

- Sum
- Avg
- Count
- Min
- Max

It is important to know the type of aggregations that the source database supports when using Direct Discovery. For example, most SQL databases support DISTINCT in any aggregation, but Google BigQuery only supports COUNT(DISTINCT ...).

Most of the QlikView Chart functions are available when using Direct Discovery fields: interactive sorting, formatting, visual clues, dimension limits, etc.

Due to the SQL syntax specific nature of Direct Discovery, pivot tables and mini charts are not supported with Direct Discovery fields.

When MEASURE fields require quoting, the quotation marks must be applied in the **Direct Query** statement, not in the Chart expression. For databases such as Oracle that control case sensitivity through the use of quoted identifiers, the column names in the Chart expressions must match the case of the column names in the database exactly.

A MEASURE field that starts with an underscore character must be enclosed by double quotation marks in a Chart expression.

The "Suppress missing" option in Charts does not work with Direct Discovery fields; it should be disabled. Once it is disabled, the "Suppress when value is null" option works as expected.



## 68 Chart Expressions

Expressions are used in charts to define calculated data, e.g. the data shown on the y-axis in a standard bar chart or line chart. Although it is often sufficient to simply click together the most basic chart expressions in the *Edit Expression Dialog* (page 757), QlikView supports far more complex chart expressions, involving functions, fields and operators. However, please note that a few differences exist between script expression and chart expressions in terms of syntax and available functions.

---

### Note!

The most important difference is the role of the aggregation functions and the use of field references. The basic rule is that any field name in a chart expression must be enclosed by exactly one aggregation function. An aggregation function can never have another expression containing an aggregation function as argument.

---

- All expressions return a number and/or a string, whichever is appropriate.
- Logical functions and operators return 0 for false, -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 or functions, return NULL.

The general syntax for a chart expression is:

```

expression ::= ( constant          |
expressionname          |
operator1 expression    |
expression operator2 expression |
function                |
aggregation 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 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 ::= aggregation { , aggregation }*

The number and types of parameters is not arbitrary. It depends on the function used.

---

General syntax for aggregation:

```
aggexpression ::= (fieldref |
operator1 aggexpression |
aggexpression operator2 aggexpression |
functioninaggr |
(aggexpression ) )
```

*fieldref* is a field name.

*functionaggr* ::= *functionname* ( *parameters2* )

Expressions and functions can thus be nested freely, as long as *fieldref* is always enclosed by exactly one aggregation function and provided the expression returns an interpretable value, QlikView will not give any error messages.

## 68.1 Alternate States and Chart Expressions

Chart and other object expressions inherit the state of the object that contains the expression.

It is possible for chart and object expressions to reference alternate states. This means that an expression, no matter where it occurs, can reference a different state than the object that contains the expression. The syntax for this reference is the same as referencing a set within Set Analysis. To understand how this works, it may be useful to review *Set Analysis (page 819)*.

One of the core features of QlikView for many years, has been the ability to use the **default state** to drive a sub-set of data on which to calculate charts and aggregations. The core algorithm takes the definition of the state in terms of Values selected per Field and determines a Set in terms of a subset of Rows per Table. Set Analysis allows this default behavior to be changed at two distinct points:

- Defining a set of data that is independent of current selections.
- Combining multiple sets through the use of mathematical operators such as Union, Intersection and Exception.

Alternate States plays a role in the first part; defining selection states from which sets can be generated. As with Set Analysis, \$ and 1 have special meanings. The **default state** is always represented by \$, while all the data, regardless of states and selections, is represented by 1.

With the introduction of Alternate States, two new syntax elements are introduced:

An expression can be based on an alternate state.

### Examples:

```
sum({[Group 1]} Sales)
```

calculates sales based on the selections in the state 'Group 1'.

```
sum({$} Sales)
```

calculates sales based on the selections in the **default state**.

Both of these expressions can exist in a single chart. This allows end-users to compare multiple states within a single object. State references within expressions override the state of the object.

Selections in a field in one state can be used as modifiers in another state.

### Example:

```
sum({[Group 1]<Region = $::Region>} Sales)
```

This syntax uses the selections in the Region field from the **default state** and modifies the state 'Group 1' with them. The effect is to keep the Region field "synchronized" between the **default state** and 'Group 1' for this expression.

## 68.2 Variable Text Expansion

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 following syntax is used:

### **$\$( \textit{variablename} )$**

$\$( \textit{variablename} )$  expands to the value in *variablename*. If *variablename* does not exist the expansion will be the empty string.

For numeric text macro expansion the following syntax is used:

### **$\$( \# \textit{variablename} )$**

$\$( \# \textit{variablename} )$  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* .

With the help of this functionality you may e.g. change the expression used in a range of charts simultaneously simply by selecting a variable value in an input box.

## 68.3 Operators

There are two types of operators in QlikView, unary operators that take only one operand, or binary operators that take two operands. Most operators are binary.

The following operators can be defined:

<b>Numeric Operators</b>	<i>page 788</i>
<b>String Operators</b>	<i>page 788</i>
<b>Logical Operators</b>	<i>page 788</i>
<b>Relational Operators</b>	<i>page 788</i>
<b>Bit Operators</b>	<i>page 789</i>

## Numeric Operators

All numeric operators use the numeric values of the operands and return a numeric value as result.

<b>+</b>	Sign for positive number (unary operator) or arithmetic addition. The binary operation returns the sum of the two operands.
<b>-</b>	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.
<b>*</b>	Arithmetic multiplication. The operation returns the product of the two operands.
<b>/</b>	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 return a string as result. The other one compares the operands and returns a boolean value to indicate match.

<b>&amp;</b>	String concatenation. The operation returns a text string, that consists of the two operand strings, one after another.
--------------	---

Example:

'abc' & 'xyz' returns 'abcxyz'

<b>like</b>	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.

<b>not</b>	Logical inverse. One of the few unary operators. The operation returns the logical inverse of the operand.
<b>and</b>	Logical and. The operation returns the logical and of the operands.
<b>or</b>	Logical or. The operation returns the logical or of the operands.
<b>Xor</b>	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 the result. All relational operators are binary.

<code>&lt;</code>	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.
<code>&lt;=</code>	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.
<code>&gt;</code>	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.
<code>&gt;=</code>	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.
<code>=</code>	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.
<code>&lt;&gt;</code>	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.
<b>precedes</b>	ASCII less than	<p>Unlike the <code>&lt;</code> operator no attempt is made to make a numeric interpretation of the argument values before the comparison. The operation returns true if the value to the left of the operator has a text representation which, in ASCII comparison, comes before the text representation of the value on the right.</p> <p><b>Example:</b></p> <pre>' 11' precedes ' 2' returns true       compare this to: ' 11' &lt; ' 2' returns false</pre>
<b>follows</b>	ASCII greater than	<p>Unlike the <code>&gt;</code> operator no attempt is made to make a numeric interpretation of the argument values before the comparison. The operation returns true if the value to the left of the operator has a text representation which, in ASCII comparison, comes after the text representation of the value on the right.</p> <p><b>Example:</b></p> <pre>' 23' follows ' 111' returns true       compare this to: ' 23' &gt; ' 111' returns false</pre>

## Bit Operators

All bit operators convert the operands to signed integers (32 bit) and return the result in the same 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.

<b>bitand</b>	Bit and.	The operation returns the logical AND of the operands performed bit by bit.
<b>bitor</b>	Bit or.	The operation returns the logical OR of the operands performed bit by bit.
<b>bitxor</b>	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.

## 68.4 Chart Aggregation functions

These functions can only be used on fields in chart expressions. The argument expression of one aggregation function must not contain another aggregation function.

The aggregation functions are:

<b>Basic Aggregation Functions</b>	<i>page 791</i>
<b>String Aggregation Functions</b>	<i>page 792</i>
<b>Counter Aggregation Functions</b>	<i>page 793</i>
<b>Statistical Aggregation Functions in Charts</b>	<i>page 794</i>
<b>Financial Aggregation Functions in Charts</b>	<i>page 803</i>
<b>Statistical Distribution Functions</b>	<i>page 805</i>
<b>Special Input Field Aggregation Functions</b>	<i>page 817</i>
<b>Advanced Aggregation</b>	<i>page 301</i>
<b>Set Analysis</b>	<i>page 819</i>
<b>Alternate States</b>	<i>page 113</i>

If the word **distinct** occurs before an *expression*, duplicates resulting from the evaluation of this *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**. 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.

*Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622). See also the examples of *Nested Aggregations and Related Issues* (page 865).

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. See also *Set Analysis* (page 819).

For a better understanding of how to use the aggregate qualifier (total), see *Examples* (page 853).

## 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).

### Examples:

```
sum(Sales)
sum(Price*Quantity)
sum(distinct Price)
sum(Sales)/sum(total Sales) returns the share within the selection
sum(Sales)/sum(total <Month> Sales) returns the share within the selection for each
Month
sum(Sales)/sum(total <Month,Grp> Sales) returns the share within the 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({1} total Sales) returns sales within the entire 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
the years 2007 and 2008, that is, the same as sum(if(Year=2007 or Year=2008, Sales))
```

```
min([{set_expression}][ distinct ] [ total [<fld {, fld}>]] expres-
sion [, 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.

### Examples:

```
min( Sales )
min( Sales, 2 )
min( Price*Quantity )
min( total Sales )
min( {1} total Sales )
```

```
max([{set_expression}][ distinct ] [ total [<fld {, fld}>]] expres-
sion [, rank])
```

Returns the numeric maximum value of *expression* or *field* iterated over the chart dimension(s). *Rank* defaults to 1 which corresponds 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:

```
max( Sales )
max( Sales, 2 )
max( Price*Quantity )
max( total Sales )
max( {1} total Sales )
```

```
only( [{set_expression}] [ distinct ] [ total [<fld {, fld}>]] expres-
sion)
```

If *expression* or *field* iterated over the chart dimension(s) contain one single value, that value is returned, else NULL is returned. **Only** can return numeric values as well as text values.

#### Examples:

```
only( Sales )
only( Price*Quantity )
only( total Salesman )
```

```
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.

**Mode** does not support the **total** qualifier.

#### Examples:

```
mode( Product )
mode( X*Y/3 )
```

```
firstsortedvalue( [{set_expression}] [ distinct ] [ total [<fld {,
fld}>]] expression [, sort_weight [, n]])
```

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.

#### 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* iterated over the chart dimension(s) contains one or more values with a string representation (any text or number), the first text value in text sort order is returned, else NULL is returned.



**Examples:**

```
MinString( Currency )
MinString( Left( abc,2 ) )
MinString( total Currency)
MinString( <X> Currency )
```

**MaxString**([{set\_expression}][ **total** [<fld {, fld}>]] *expression*)

If *expression* iterated over the chart dimension(s) contains one or more values with a string representation (any text or number), the last text value in text sort order is returned, else NULL is returned.

**Examples:**

```
MaxString( Currency )
MaxString( Left( abc,2 ) )
MaxString( total Currency)
MaxString( total <X> Currency )
```

**concat**([{set\_expression}] [ **distinct** ] [ **total** [<fld {, fld}>]] *expression* [, *delimiter* [, *sort\_weight*]])

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.

**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). For this function it is allowed to use the **distinct** qualifier in combination with the **total** qualifier. This combination is not valid for any other aggregation functions.

**Examples:**

```
count(Sales)
count(Price*Quantity)
count(distingct Price)
count(Sales)/count(total Sales) returns sales within the selection
```

**NumericCount**([{set\_expression}][ **distinct** ] [ **total** [<fld {, fld}>]] *expression*)

Returns the aggregated numeric count of values from *expression* or *field* iterated over the chart dimension(s).

**Examples:**

```
NumericCount(Sales)
```

```
NumericCount(Price*Quantity)
NumericCount(distinct Price)
NumericCount(Sales)/NumericCount(total Sales)
NumericCount(Sales)/NumericCount({1} total Sales) returns sales within the total document
```

```
TextCount([{set_expression}][ distinct ] [ total [<fld {, fld}>]] expression)
```

Returns the aggregated text count of values from *expression* or *field* iterated over the chart dimension(s).

**Examples:**

```
TextCount(Sales)
TextCount(Price*Quantity)
TextCount(distinct Price)
TextCount(Sales)/TextCount(total Sales) returns sales within the selection
TextCount(Sales)/TextCount({1} total Sales) returns sales within the total document
```

```
NullCount([{set_expression}][ distinct ] [ total [<fld {, fld}>]] expression)
```

Returns the aggregated count of NULL values from *expression* or *field* iterated over the chart dimension(s).

**Examples:**

```
NullCount(Sales)
NullCount(Price*Quantity)
NullCount(distinct Price)
NullCount(Sales)/NullCount(total Sales) returns sales within the selection
NullCount(Sales)/NullCount({1} total Sales) returns sales within the total document
```

```
MissingCount([{set_expression}][ distinct ] [ total [<fld {, fld}>]] expression)
```

Returns the aggregated count of missing values from *expression* or *field* iterated over the chart dimension(s). Missing values are all non-numeric values.

**Examples:**

```
MissingCount(Sales)
MissingCount(if(Price>10, Price, 'invalid'))
MissingCount(distinct Price)
MissingCount(Sales)/MissingCount(total Sales) returns sales within the selection
MissingCount(Sales)/MissingCount({1} total Sales) returns sales within the total document
```

-

## Statistical Aggregation Functions in Charts

```
avg([{set_expression}][ distinct ] [ total [<fld {, fld } >]] expression)
```

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 *Advanced Aggregation (page 301)* function in combination with calculated dimensions, see *Add calculated dimension... (page 622)*.

If the word **distinct** occurs before the function arguments, duplicates resulting from the evaluation of the function arguments 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 **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.

### Examples:

```
avg(Sales)
avg(X*Y/3)
avg(distinct Price)
avg(total Sales)
avg({1} total Sales)
```

```
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.

This function has the same limitations for nested aggregation as the *avg({set\_expression} [distinct] [total [<fld {,fld}>]] expression) (page 794)* function. The **sterr** function supports *Set Analysis (page 819)* and the **total** qualifier in the same way as the *avg({set\_expression} [distinct] [total [<fld {,fld}>]] expression) (page 794)* function.

### 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 *y-expression* and *x-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.

This function has the same limitations for nested aggregation as the *avg({set\_expression} [distinct] [total [<fld {,fld}>]] expression) (page 794)* function. The **steyx** function supports *Set Analysis (page 819)* and the **total** qualifier in the same way as the *avg({set\_expression} [distinct] [total [<fld {,fld}>]] expression) (page 794)* function.

### 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.

This function has the same limitations for nested aggregation as the *avg*([{set\_expression}] [ **distinct** ] [ **total** [<fld {,fld}>]] expression) (page 794) function. The **linest\_m** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the *avg*([{set\_expression}] [ **distinct** ] [ **total** [<fld {,fld}>]] expression) (page 794) function.

#### 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.

This function has the same limitations for nested aggregation as the *avg*([{set\_expression}] [ **distinct** ] [ **total** [<fld {,fld}>]] expression) (page 794) function. The **linest\_b** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the *avg*([{set\_expression}] [ **distinct** ] [ **total** [<fld {,fld}>]] expression) (page 794) function.

#### 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.

This function has the same limitations for nested aggregation as the `avg({set_expression} [distinct] [total [<fld {,fld}>]] expression)` (page 794) function. The **linest\_r2** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg({set_expression} [distinct] [total [<fld {,fld}>]] expression)` (page 794) function.

### 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.

This function has the same limitations for nested aggregation as the `avg({set_expression} [distinct] [total [<fld {,fld}>]] expression)` (page 794) function. The **linest\_sem** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg({set_expression} [distinct] [total [<fld {,fld}>]] expression)` (page 794) function.

### 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.

This function has the same limitations for nested aggregation as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function. The **linest\_seb** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function.

### 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.

This function has the same limitations for nested aggregation as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function. The **linest\_sey** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function.

### 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.

This function has the same limitations for nested aggregation as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function. The **linest\_df** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function.

### 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.

This function has the same limitations for nested aggregation as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function. The **linest\_f** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg([set_expression] [distinct] [total [<fld {,fld}>]] expression)` (page 794) function.

### 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_ssreg ( [ {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.

This function has the same limitations for nested aggregation as the `avg({set_expression} [distinct] [total [ <fld {,fld}> ] ] expression)` (page 794) function. The **linest\_ssreg** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg({set_expression} [distinct] [total [ <fld {,fld}> ] ] expression)` (page 794) function.

### 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.

This function has the same limitations for nested aggregation as the `avg({set_expression} [distinct] [total [ <fld {,fld}> ] ] expression)` (page 794) function. The **linest\_ssresid** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg({set_expression} [distinct] [total [ <fld {,fld}> ] ] expression)` (page 794) function.

### 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 )
```



```
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 *range-correl(x-value , y-value { , x-value , y-value})* (page 836) function.

*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 *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622).

This function has the same limitations for nested aggregation as the *avg({set\_expression} [ distinct ] [ total [ <fld { , fld } > ] ] expression)* (page 794) function. The **correl** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the *avg({set\_expression} [ distinct ] [ total [ <fld { , fld } > ] ] expression)* (page 794) function.

### Examples:

```
correl( Payments, Dates )
correl( A/B, X*Y/3 )
correl( total Payments, Dates )
correl( total <Grp> Payments, Dates)
```

```
median ( [ {set_expression} ] [ distinct ] [ total [ <fld { , fld } > ] ] expression )
```

Returns the aggregated median of *expression* iterated over the chart dimension(s).

This function has the same limitations for nested aggregation as the *avg({set\_expression} [ distinct ] [ total [ <fld { , fld } > ] ] expression)* (page 794) function. The **median** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the *avg({set\_expression} [ distinct ] [ total [ <fld { , fld } > ] ] expression)* (page 794) function.

### Examples:

```
median( X )
median( X*Y/3 )
median( total X )
median( total <Group> Price )
```

```
stdev( [ {set_expression} ] [ distinct ] [ total [ <fld { , fld } > ] ] expression)
```

Returns the aggregated standard deviation of *expression* or *field* iterated over the chart dimension(s).

This function has the same limitations for nested aggregation as the *avg({set\_expression} [ distinct ] [ total [ <fld { , fld } > ] ] expression)* (page 794) function. The **stdev** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the *avg({set\_expression} [ distinct ] [ total [ <fld { , fld } > ] ] expression)* (page 794) function.

### Examples:

```
stdev(Sales)
stdev(X*Y/3)
stdev(distinct Price)
stdev(total Sales)
```

```
stdev({1} total Sales)
```

```
skew([{set_expression}] [ distinct ] [ total [<fld { , fld } >] ]  
expression )
```

Returns the aggregated skewness of *expression* or *field* iterated over the chart dimension(s).

This function has the same limitations for nested aggregation as the `avg`([{set\_expression}] [ **distinct** ] [ **total** [<fld { , fld } >]] *expression*) (page 794) function. The **skew** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg`([{set\_expression}] [ **distinct** ] [ **total** [<fld { , fld } >]] *expression*) (page 794) function.

### Examples:

```
skew(Sales)  
skew(X'Y/3)  
skew(distinct Price)  
skew(total Sales)  
skew({1} total Sales)
```

```
kurtosis([{set_expression}][ distinct ] [ total [<fld { , fld } >]  
] expression)
```

Returns the aggregated kurtosis of *expression* or *field* iterated over the chart dimension(s).

This function has the same limitations for nested aggregation as the `avg`([{set\_expression}] [ **distinct** ] [ **total** [<fld { , fld } >]] *expression*) (page 794) function. The **kurtosis** function supports *Set Analysis* (page 819) and the **total** qualifier in the same way as the `avg`([{set\_expression}] [ **distinct** ] [ **total** [<fld { , fld } >]] *expression*) (page 794) function.

### Examples:

```
kurtosis(Sales)  
kurtosis(X'Y/3)  
kurtosis(distinct Price)  
kurtosis(total Sales)  
kurtosis({1} total Sales)
```

```
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 *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622).

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.

### Examples:

```
fractile( Sales, 0.75 )
fractile( X*Y/3, 0.9 )
fractile( total Price, 0.25 )
fractile( total <Group> Price )
```

## Financial Aggregation Functions in Charts

**irr**([[set\_expression] **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(value {,value})* (page 323) function.

*Expression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622).

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.

### Examples:

```
irr( Payments )
irr( X*Y/3 )
irr( total Payments )
irr( total <Grp> Payments )
```

**xirr** ([set\_expression] [ **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 to be disregarded. See also the *rangexirr(value, date {,value, date})* (page 324) function.

*Valueexpression* and *dateexpression* must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622).

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.

### Examples:

```
xirr( Payments, Dates )
xirr( A/B, X*Y/3 )
xirr( total Payments, Dates )
xirr( total <Grp> Payments, Dates )
```

```
npv ( [set_expression] [ 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.

Rate and expression must not contain aggregation functions, unless these inner aggregations contain the **total** qualifier. For more advanced nested aggregations, please use the *Advanced Aggregation (page 301)* function in combination with calculated dimensions, see *Add calculated dimension... (page 622)*.

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.

### 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] [ 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 to be disregarded. See also the *rangexnpv(rate, value, date {,value, date} )* (page 324) function.

*Rate, valueexpression* and *dateexpression* must not contain aggregation functions, unless these inner aggregations contain the total or all qualifiers. For more advanced nested aggregations, please use the *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622).

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 **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.

### 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 in Charts

The following three functions apply to chi2-tests.

```
chi2test_p ([set_expression] [ total [<fld {,fld}>] col, row,
observed_value [, expected_value])
```

returns the aggregated chi<sup>2</sup>-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 *Advanced Aggregation* (page 301) function in combination with calculated dimensions, see *Add calculated dimension...* (page 622).

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.

### 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* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Examples:

```
chi2test_chi2( Grp, Grade, Count )
chi2test_chi2( Gender, Description, Observed, Expected )
```

The following eight functions apply to 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.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) function for descriptions of the other arguments.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) and *TTest\_conf* ([set\_expression][ total [<fld {fld}>] group, value [, sig = 0.025 [, eq\_var = true]]) (page 808) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] group, value [, eq\_var = true]) (page 806) and *TTest\_conf* ([set\_expression][ total [<fld {fld}>] group, value [, sig = 0.025 [, eq\_var = true]]) (page 808) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Examples:

```
ttest_upper( Group, Value )
ttest_upper( Group, Value, false )
```

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 ([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.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] weight, group, value [, eq\_var = true]) (page 808) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] weight, group, value [, eq\_var = true]) (page 808) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] weight, group, value [, eq\_var = true]) (page 808) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, group, value [, eq\_var = true]) (page 808) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, group, value [, eq\_var = true]) (page 808) function for descriptions of the other arguments.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, group, value [, eq\_var = true]) (page 808) and *TTestw\_conf* ([set\_expression][ total [<fld {fld}>] weight, group, value [, sig = 0.025 [, eq\_var = true]]) (page 810) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, group,

*value* [, *eq\_var* = true]) (page 808) and *TTestw\_conf* ([*set\_expression*][ *total* [<fld {fld}>] *weight*, *group*, *value* [, *sig* = 0.025 [, *eq\_var* = true]]) (page 810) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function.

### 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.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function.

### 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* ([*set\_expression*][ *total* [<fld {fld}>] *value*) (page 811) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function.

### 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* ([*set\_expression*][ *total* [<fld {fld}>] *value*) (page 811) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([*set\_expression*][ *total* [<fld {fld}>] *col*, *row*, *observed\_value* [, *expected\_value*]) (page 805) function.

### 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* ([*set\_expression*][ *total* [<fld {fld}>] *value*) (page 811) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] value) (page 811) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] value) (page 811) function for descriptions of the other arguments.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][total [<fld {fld}>] value) (page 811) and *TTest1\_conf* ([set\_expression][total [<fld {fld}>] value [, sig = 0.025 ]) (page 812) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] value) (page 811) and *TTest1\_conf* ([set\_expression][ total [<fld {fld}>] value [, sig = 0.025 ]) (page 812) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Example:

```
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* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Example:

```
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* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) function for descriptions of the other arguments.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) and **TTest1w\_conf** functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Examples:

```
ttest1w_lower( Weight, Value )
```

```
ttest1w_lower( Weight, Value, 0.005 )
```

---

```
TTestlw_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 *TTestlw\_t* ([set\_expression][ total [<fld {fld}>] weight, value) (page 813) and **Ttestlw\_conf** functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Examples:

```
ttestlw_upper( Weight, Value )
ttestlw_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.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] value [, sigma]) (page 815) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {fld}>] value [, sigma]) (page 815) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {,fld}>] value [, sigma]) (page 815) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {,fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {,fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### 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* ([set\_expression][ total [<fld {,fld}>] value [, sigma]) (page 815) function for descriptions of the other arguments.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {,fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {,fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Example:

```
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.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][ total [<fld {,fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][ total [<fld {,fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

### Example:

```
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* ([set\_expression][ total [<fld {,fld}>] weight, value [, sigma]) (page 816) function for argument descriptions.



The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

#### Example:

```
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* ([set\_expression][total [<fld {fld}>] weight, value [, sigma]) (page 816) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

#### Example:

```
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* ([set\_expression][total [<fld {fld}>] weight, value [, sigma]) (page 816) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

#### Example:

```
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* ([set\_expression][total [<fld {fld}>] weight, value [, sigma]) (page 816) function for descriptions of the other arguments.

The same limitations with regard to nested aggregation as those listed under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function apply. This function supports the **total** qualifier in the same manner as described under the *chi2test\_p* ([set\_expression][total [<fld {fld}>] col, row, observed\_value [, expected\_value]) (page 805) function.

#### Examples:

```
ztestw_conf( Weight, Value-TestValue)
```

## Special Input Field Aggregation Functions

There are some special aggregation functions for input fields:

*inputsum* (inputfield[, distribution\_mode][set\_expression]) (page 818)

*inputavg*(inputfield [, distribution\_mode][set\_expression]) (page 818)

**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 on 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.

The *distribution\_mode* parameter can have the following values:

- '+' Default mode. Equal parts of the change are 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 on 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.

The *distribution\_mode* parameter can have the following values:

- '+' Default mode. Equal parts of the change are 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' )
```

## Set Analysis

Aggregation functions by default aggregate over the current selection of field values. The current selection can be referred to as a set of field values. You can define other sets of field values and use them in your charts instead of the current selection. For example, in a dashboard you may want to show market share a product across all regions irrespective of the current selections.

Defining a set of field values is referred to as defining a set expression and using set expressions to analyze data is referred to as set analysis.

---

### Note!

Set expressions are available for charts only, and not in scripts.

---

Set expressions always begin and end in curly brackets. For example, `sum({1} Sales)` where `{1}` is a set expression.

## Building a set expression

In a set expression you must always identify how the set of field values you are defining relates to the field or expression you are evaluating. For example, are you evaluating the complete set of field values or the inverse of the current selection? After you have identified this relationship you can modify the selection of values within the field (this is optional).

In short, a set expression includes an identifier and an optional modifier. Modifiers are separated from identifiers by angled brackets as follows:

```
{set_identifier<set_modifier>}
```

You can use operators on both identifiers and modifiers to manipulate field relationships and selections. Additionally, QlikView enables you to combine modifiers with dollar-sign expansions, advanced searches and implicit field value definitions as described in the following topics.

Examples:

To understand the basics of set expression, let's look at a simple use case. We want to build a dashboard showing the following charts of value sales in the USA:

1. Total value sales in the USA by product group irrespective of current selection
2. Value sales in the USA by product group given the current selection

Our sales data is global but is divided in Region. Product groups are found in the ProductGroup field while value sales values are found in the Sales field.

In normal circumstances you could build one chart with the dimension ProductGroup and the measure `sum(Sales)`. User selections of Region and ProductGroup would then determine what is shown. However here

we want chart (1) above to always show the same region and selections and chart (2) to always show the same region. Let's express what we want to see in each chart in terms of a set expression:

1. set\_expression= {all of Sales <for RegionUSA>}
2. set\_expression= {current selection of Sales< for RegionUSA>}2. set\_expression= {current selection of Sales< for RegionUSA>}

Given that the:

qualifier for "all" is 1

qualifier for current selection is \$

syntax for modifiers in this case is set\_modifier = <field\_name={field\_value,[field\_value]}

we get the following measures for our charts using set expressions:

1. sum({1<Region={USA}>} Sales)
2. sum({\$<Region={USA}>} Sales)

## Set Identifiers

Set identifiers define the relationship between the set expression and the field values or expression that is being evaluated.

Set identifiers can be combined using set operators.

Identifier	Description
1	Represents the full set of all the records in the application.
\$	Represents the records of the current selection. The set expression {\$} is thus the equivalent of not stating a set expression.
	<b>Note!</b> {1-\$} is all the more interesting as it defines the inverse of the current selection, that is, everything that the current selection excludes.
\$N	Selections from the Back stack can be used as set identifiers, by use of the dollar symbol: \$1 represents the previous selection, that is, it is equivalent to pressing the Back button. Any unsigned integer can be used in the Back notation. \$0 represents the current selection.
\$_N	Selections from the Forward stack can be used as set identifiers, by use of the dollar symbol: \$_1 represents one step forward, that is the equivalent to pressing the Forward button. Any unsigned integer can be used in the Forward notation. \$0 represents the current selection.
bookmark_id   bookmark_name	Server and app bookmarks can be used as set identifiers. Either the bookmark ID or the bookmark name can be used. For example <b>BM01</b> or <b>MyBookMark</b> . Only the selection part of a bookmark is used. The 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 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

There are several set operators that can be used in set expressions. All set operators use sets as operands and return a set as result.

The order of precedence is:

1. Unary minus (complement)
2. Intersection and Symmetric difference
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 because the set operators do not commute. For example,  $A+(B-C)$  is different from  $(A+B)-C$  which in turn is different from  $(A-C)+B$ .

---

### 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  $\text{sum}(\{*\text{BM01}\}\text{Quantity}*\text{Price})$  should be avoided.

---

### Arguments:

Operator	Description
+	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 to the first but not the other 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.

### 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.

## 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 >. For example: <Year={2007,+2008},Region={US}>. Field names and field values can be quoted as usual, e.g. <[Sales Region]='West coast', 'South America'>.

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 modifier must be written immediately after the set identifier, for examples {\$<Year = {2007, 2008}>}. When used on its own, it is interpreted as a modification of the current selection.

There are several ways to define the selection as described in the following.

### Based on another field

A simple case is a selection based on the selected values of another field, for example <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.

### Based on element sets ( a field value list in the modifier)

The most common case is a selection based on a field value list enclosed in curly brackets, the values separated by commas, for example <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, for example <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 for example <Product = {}> or implicitly for example <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, for example **TransactionID**.

### Forced exclusion

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.

#### Examples:

**sum( {1<Region= {US} >} Sales )**

returns the sales for 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 current selection, but with new selections both in “Year” and in “Region”.

---

**sum( {\$<Ingredient = {"\*garlic\*"}>} Sales )**

returns the sales for current selection, but with a forced exclusion of all Ingredients containing the string 'garlic'.

**sum( {\$<Year = {"2\*"}>} Sales )**

returns the sales for 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 also the 1980:s are 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 `<Year = {"20*", 1997} - {2000}>` will select all years beginning with "20" in addition to "1997", *except* for "2000".

### **Examples:**

**sum( {\$<Product = Product + {OurProduct1} - {OurProduct2} >} 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.

**sum( {\$<Year = Year + ({"20\*",1997} - {2000}) >} 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.

**sum( {\$<Year = (Year + {"20\*",1997}) - {2000} >} Sales )**

returns almost the same as above, but here 2000 will be excluded, also if it initially is included in the current selection. The example shows the importance of sometimes using brackets to define an order of precedence.

**sum( {\$<Year = {"\*"} - {2000}, Product = {"\*bearing\*"} >} 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, e.g. you may want a modifier `<Year = Year + {2007, 2008}>`. A short and equivalent way to write this is `<Year += {2007, 2008}>`, i.e. the assignment operator implicitly defines a union. Also implicit intersections, exclusions and symmetric differences can be defined using `"*="`, `"-="` and `"/="`.

### **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 Advanced Searches

Advanced searches using wildcards and aggregations can be used to define sets.

### Examples:

**sum( {\$-1<Product = {"\*Internal\*", "\*Domestic\*"}>} Sales )**  
returns the sales for 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 current selection, but with a new selection in the "Customer" field: only customers who during 2007 had a total sales of more than 1000000.

## 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 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 current selection. Here, a dollar-sign expansion is used to calculate previous year.

## 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 "

```

## 68.5 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

```

## 68.6 Other Functions

QlikView chart expressions support close to one hundred standard functions, apart from the aggregation functions described above. Some of the functions below are the same as for Scripts, see Script section for details. The functions have been divided into groups as follows:

<b>Chart General Numeric Functions</b>	<i>page 826</i>
<b>Chart Range Functions</b>	<i>page 830</i>
<b>Exponential and Logarithmic Functions</b>	<i>page 325</i>
<b>Trigonometric and Hyperbolic Functions</b>	<i>page 325</i>

<b>Statistical Distribution Functions</b>	<i>page 326</i>
<b>Financial Functions</b>	<i>page 328</i>
<b>Mathematical Constants and Parameter-Free Functions</b>	<i>page 330</i>
<b>String Functions</b>	<i>page 333</i>
<b>Chart Inter Record Functions</b>	<i>page 836</i>
<b>Document Functions</b>	<i>page 350</i>
<b>Chart Ranking Functions</b>	<i>page 843</i>
<b>Conditional Functions</b>	<i>page 340</i>
<b>Logical Functions</b>	<i>page 341</i>
<b>NULL Functions</b>	<i>page 342</i>
<b>System Functions</b>	<i>page 342</i>
<b>Date and Time Functions</b>	<i>page 350</i>
<b>Interpretation Functions</b>	<i>page 847</i>
<b>Formatting Functions</b>	<i>page 370</i>
<b>Color Functions</b>	<i>page 375</i>

## Chart General Numeric Functions

In the functions below, the parameters are expressions where  $x$  should be interpreted as a real valued number.

### **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 non-negative remainder of an integer division.

#### **Examples:**

- mod**( 7,2 ) returns 1
- mod**( 7.5,2 ) returns NULL
- 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( 7.5,2 )** returns 1.5  
**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* [ , *base* [ , *offset* ] ] )**

Rounding of *x* upwards or downwards to the nearest multiple of *base* with an offset of *offset*. The result is a number. If *x* is exactly in the middle of an interval, it is rounded upwards.

#### **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

#### **numsum ( expr1 [ , expr2, ... exprN ] )**

Returns the numeric sum of 1 to N arguments. As opposed to the + operator, **numsum** will treat all non-numeric values as 0.

#### **Examples:**

**numsum( 1,2,4 )** returns 7  
**numsum( 1,'xyz' )** returns 1  
**numsum( null() )** returns 0

---

#### **Note!**

The **numsum** function is now obsolete and is superseded by the *rangesum(expr1 [ , expr2, ... exprN ])* (page 831) function.

---

#### **numcount (expr1 [ , expr2, ... exprN ] )**

Returns the number of numeric values found in 1 to N arguments

#### **Examples:**

**numcount( 1,2,4, )** returns 3  
**numcount( 2,xyz )** returns 1  
**numcount( null() )** returns 0

---

#### **Note!**

The **numcount** function is now obsolete and is superseded by the *rangecount(expr1 [ , expr2, ... exprN ])* (page 831) function.

---

#### **numavg (expr1 [ , expr2, ... exprN ] )**

Returns the numeric average of 1 to N arguments. If no numeric value is found, NULL is returned.

#### **Examples:**

**numavg( 1,2,4 )** returns 2.33333333  
**numavg( 1,'xyz' )** returns 1

---

---

**numavg( null() 'abc' )** returns NULL

---

**Note!**

The **numavg** function is now obsolete and is superseded by the *rangeavg(expr1 [ , expr2, ... exprN ])* (page 831) function.

---

**nummin( expr1 [ , expr2, ... exprN ] )**

Returns the lowest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

**Examples:**

**nummin( 1,2,4 )** returns 1

**nummin( 1,'xyz' )** returns 1

**nummin( null() 'abc' )** returns NULL

---

**Note!**

The **nummin** function is now obsolete and is superseded by the *rangemin(expr1 [ , expr2, ... exprN ])* (page 831) function.

---

**nummax( expr1 [ , expr2, ... exprN ] )**

Returns the highest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

**Examples:**

**nummax( 1,2,4 )** returns 4

**nummax( 1,'xyz' )** returns 1

**nummax( null() 'abc' )** returns NULL

---

**Note!**

The **nummax** function is now obsolete and is superseded by the *rangemax(expr1 [ , expr2, ... exprN ])* (page 832) function.

---

**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 selected is insignificant. Non-integer items will be truncated.

**Example:**

How many combinations of 7 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 selected 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

**sign** (*x*)

Returns 1, 0 or -1 depending on whether *x* is a 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

**bitcount** (*i* )

Returns the number of set bits in *i*. Where *i* is interpreted as a signed 32-bit integer.

**Examples:**

**bitcount ( 3 )** returns 2

**bitcount ( -1 )** returns 32

**bitcount ( *n* )+bitcount ( **bitnot** *n* )** returns 32 (as long as *n* is a valid integer)

## Chart Range Functions

Range functions will replace the following *Chart General Numeric Functions (page 826)*: **numsum**, **numavg**, **numcount**, **nummin** and **nummax**, which should now be regarded as obsolete.

**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.

The argument expressions of this function may contain *Chart Inter Record Functions (page 836)* with a third optional parameter, which in themselves return a range of values.

**Examples:**

<b>rangesum</b> (1,2,4)	returns 7
<b>rangesum</b> (1,'xyz')	returns 1
<b>rangesum</b> (null( ))	returns 0
<b>rangesum</b> (above(count(x),-1,3))	returns the sum of the three results of the <b>count(x)</b> 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 *Chart Inter Record Functions (page 836)* with a third optional parameter, which in themselves return a range of values.

**Examples:**

<b>rangeavg</b> (1,2,4)	returns 2.33333333
<b>rangeavg</b> (1,'xyz')	returns 1
<b>rangeavg</b> (null( ), 'abc')	returns NULL
<b>rangeavg</b> (above(count(x),0,3))	returns a sliding average of the result of the inner <b>count(x)</b> expression calculated on 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 *Chart Inter Record Functions (page 836)* with a third optional parameter, which in themselves return a range of values.

**Examples:**

<b>rangecount</b> (1,2,4)	returns 3
<b>rangecount</b> (2,'xyz')	returns 2
<b>rangecount</b> (null( ))	returns 0
<b>rangecount</b> (above(sum(x),-1,3))	returns the number of values within the three results of the <b>sum(x)</b> 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.

**Examples:**

<b>rangemin (1,2,4)</b>	returns 1
<b>rangemin (1,'xyz')</b>	returns 1
<b>rangemin (null( ), 'abc')</b>	returns NULL
<b>rangemin (above(sum(x),-1,3))</b>	returns the lowest of the three results of the <b>sum(x)</b> 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.

**Examples:**

<b>rangemax (1,2,4)</b>	returns 4
<b>rangemax (1,'xyz')</b>	returns 1
<b>rangemax (null( ), 'abc')</b>	returns NULL
<b>rangemax (above(sum(x),-1,3))</b>	returns the highest of the three results of the <b>sum(x)</b> 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 *Chart Inter Record Functions (page 836)* with a third optional parameter, which in themselves return a range of values.

**Examples:**

<b>rangestdev (1,2,4)</b>	returns 1.5275252316519
<b>rangestdev (null( ))</b>	returns NULL
<b>rangestdev (above(count(x),0,3))</b>	returns a sliding standard deviation of the result of the inner <b>count(x)</b> 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 *Chart Inter Record Functions (page 836)* with a third optional parameter, which in themselves return a range of values.

**Examples:**

<b>rangeskew (1,2,4)</b>	returns 0.93521952958283
<b>rangeskew (above(count(x),0,3))</b>	returns a sliding skewness of the result of the inner <b>count(x)</b> 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 *Chart Inter Record Functions (page 836)* 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 fractile of numeric values found within a range of 1 to N arguments. The argument expressions of this function may contain *Chart Inter Record Functions (page 836)* with a third optional parameter, which in themselves return a range of values.

**Examples:**

**rangefractile (0.24,1,2,4,6)** returns 1

**rangefractile (0.5,1,2,3,4,6)** returns 3

**rangefractile (0.5,1,2,5,6)** returns 3.5

**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 *Chart Inter Record Functions (page 836)* 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 *Chart Inter Record Functions (page 836)* 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 NULL values found within a range of 1 to N arguments.

The argument expressions of this function may contain *Chart Inter Record Functions (page 836)* 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 *Chart Inter Record Functions (page 836)* 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.

**rangeminstring** (*expr1 [ , expr2, ... exprN ]*)

Returns the first value in text sort order found among 1 to N arguments.

The argument expressions of this function may contain *Chart Inter Record Functions (page 836)* 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 value in text sort order found among 1 to N arguments.

The argument expressions of this function may contain *Chart Inter Record Functions (page 836)* 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.

**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 *Chart Inter Record Functions (page 836)* 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 *Chart Inter Record Functions* (page 836) with a third optional parameter, which in themselves return a range of values.

**Examples:**

<b>rangeonly</b> (1,2,4)	returns NULL
<b>rangeonly</b> (1,'xyz')	returns NULL
<b>rangeonly</b> (null( ), 'abc')	returns 'abc'
<b>rangeonly</b> (above(sum(x),-1,3))	uses the three results of the <b>sum(x)</b> 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.

**rangecorrel** (*x-value* , *y-value* { , *x-value* , *y-value* })

Returns the correlation coefficient for a series of coordinates.

*X-value* is a single value or a range of values as returned by an *Chart Inter Record Functions* (page 836) with a third optional parameter.

*Y-value* is a single value or a range of values as returned by an inter-record function with a third optional parameter. 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 be calculated. 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)))

## Chart Inter Record Functions

These functions can only be used in chart expressions.

**Note!**

Sorting on y-values in charts or sorting by expression columns in straight tables is not allowed when Chart Inter Record functions are used in any of the chart's expressions. These sort alternatives are therefore automatically disabled.

Suppression of zero values is automatically disabled when these functions are used.

See *Examples of Chart Inter Record Functions* (page 859).

**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.

---

Note: This function will only work with distinct field values.

---

**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' )
```

**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 table it will return the number of dimensions in all rows except the total, which will be 0.

See *Examples of Chart Inter Record Functions (page 859)*.

**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.

See *Examples of Chart Inter Record Functions (page 859)*.

**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 bitmap charts, in the chart's straight table equivalent (Actually all QlikView charts have a straight table equivalent with the exception of the pivot table which has a more complex structure.).

On the first row of a column segment a NULL value will be returned, as there is no row above this one.

If the chart is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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 left to right. For other chart types this can be manipulated in the **Chart Properties: Sort** dialog.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* to rows further up the current row. A negative *offset* number will actually make the **above** function equivalent to a **below** function with the corresponding positive *offset* number. Specifying an *offset* of 0 will evaluate the *expression* on the current row. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of *n* table rows counting upwards from the original cell. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

**Examples:**

```
sum( Sales ) / above( sum( Sales ) )
above( sum( Sales ), 2 )
above( total sum( Sales ) )
```

**rangeavg** returns an average of the three results of the sum(x) function evaluated on the  
**(above(sum(x),1,3))** three rows immediately above the current row.

**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 bitmap 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 this one.

If the chart is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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 left to right. For other chart types this can be manipulated in the **Chart Properties: Sort** dialog.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* to rows further down the current row. A negative *offset* number will actually make the **below** function equivalent to a **above** function with the corresponding positive *offset* number. Specifying an *offset* of 0 will evaluate the *expression* on the current row. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of *n* table rows counting downwards from the original cell. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

**Examples:**

```
below( sum( Sales ) )
below( sum( Sales ), 2 )
below( total sum( Sales ) )
```

**rangeavg** returns an average of the three results of the sum(x) function evaluated on the  
**(below(sum(x),1,3))** three rows immediately below the current row.

**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 current column segment in a table or, in the case of bitmap charts, in the chart's straight table equivalent.

If the chart is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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 left to right. For other chart types this can be manipulated in the **Chart Properties: Sort** dialog.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* down to rows below the top row. A negative *offset* number will actually make the **top** function equivalent to a **bottom** function with the corresponding positive *offset* number. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of the first *n* rows of the current column segment. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

### 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 current column segment.

### **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 current column segment in a table or, in the case of bitmap charts, in the chart's straight table equivalent.

If the chart is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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 left to right. For other chart types this can be manipulated in the **Chart Properties: Sort** dialog.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* up to rows above the bottom row. A negative *offset* number will actually make the **bottom** function equivalent to a **top** function with the corresponding positive *offset* number. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of the last *n* rows of the current column segment. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

### 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 current column segment.

### **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.

On the first column of a row segment a NULL value will be returned, as there is no column before this one. If the table is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* to columns further to the left of the current column. A negative *offset* number will actually make the **before** function equivalent to a **after** function with the corresponding positive *offset* number. Specifying an *offset* of 0 will evaluate the *expression* on the current column. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of *n* table columns counting to the left from the original cell. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

### 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.

```
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.

On the last column of a row segment a NULL value will be returned, as there is no column after this one.

If the table is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* to columns further to the right of the current column. A negative *offset* number will actually make the **after** function equivalent to a **before** function with the corresponding positive *offset* number. Specifying an *offset* of 0 will evaluate the *expression* on the current column. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of *n* table columns counting to the right from the original cell. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

### Examples:

```
after( sum( Sales ) )
after( sum( Sales ), 2 )
after( total sum( Sales ) )
```



**rangeavg** 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.  
**(after(sum(x),1,3))**

**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 **total** qualifier, 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.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* to columns further to the right of the first column. A negative *offset* number will actually make the **first** function equivalent to a **last** function with the corresponding positive *offset* number. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of the first *n* columns of the current row segment. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

#### Examples:

```
first( sum( Sales ) )
first( sum( Sales ), 2 )
first( total sum( Sales ) )
```

**rangeavg** returns an average of the results of the sum(x) function evaluated on the five leftmost columns of the current row segment.  
**(first(sum(x),1,5))**

**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 **total** qualifier, 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.

Specifying an *offset* greater than 1 lets you move the evaluation of *expression* to columns further to the left of the last column. A negative *offset* number will actually make the **last** function equivalent to a **first** function with the corresponding positive *offset* number. Recursive calls will return NULL.

By specifying a third parameter *n* greater than 1, the function will return not one but a range of *n* values, one for each of the last *n* columns of the current row segment. In this form, the function can be used as an argument to any of the special *Chart Range Functions* (page 830).

#### Examples:

```
last( sum( Sales ) )
last( sum( Sales ), 2 )
```

```
last( total sum( Sales )
```

**rangeavg** returns an average of the results of the sum(x) function evaluated on the five  
**(last(sum(x),1,5))** rightmost columns of the current row segment.

### **RowNo ( [total ] )**

Returns the number of the current row within the current column segment in a table or, in the case of bitmap charts, within the chart's straight table equivalent. The first row is number 1.

If the table is one-dimensional or if the qualifier **total** is used 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 left to right. For other chart types this can be manipulated in the **Chart**

**Properties: Sort** dialog.

#### **Example:**

```
if( RowNo( )=1, 0, sum( Sales ) / above( sum( Sales )))
```

### **ColumnNo ( [total ] )**

Returns the number of the current column within the current row segment in a pivot table. The first column is number 1.

If the pivot table is one-dimensional or if the qualifier **total** is used 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.

#### **Example:**

```
if( ColumnNo( )=1, 0, sum( Sales ) / before( sum( Sales )))
```

### **NoOfRows ( [total ] )**

Returns the number of rows in the current column segment in a table or, in the case of bitmap charts, in the chart's straight table equivalent.

If the chart is one-dimensional or if the qualifier **total** is used 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 left to right. For other chart types this can be manipulated in the **Chart**

**Properties: Sort** dialog.

#### **Example:**

```
if( RowNo( )=NoOfRows( ), 0, after( sum( Sales )))
```

### **NoOfColumns ( [total ] )**

Returns the number of columns in the current row segment in a pivot table.

---

If the pivot table is one-dimensional or if the qualifier **total** is used 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 dimension in 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.

**Example:**

```
if( ColumnNo( )=NoOfColumns( ), 0, after( sum( Sales )))
```

**fieldvaluecount(*fieldname* )**

Returns the number of distinct values in a field. *Fieldname* must be given as a string (e.g. a quoted literal).

**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.
```

## Chart Ranking Functions

These functions can only be used in chart expressions.

**Note!**

Suppression of zero values is automatically disabled when these functions are used. NULL values are disregarded.

*Examples of Chart Ranking Functions (page 857)*

**rank([ *total* ] *expression* [ , *mode* [ , *format* ] ])**

Evaluates *expression*, compares the result with the result of the other rows containing the current column segment and returns the ranking of the current row within the segment. For bitmap charts, the current column segment is defined as it appears in the chart's straight table equivalent (Actually all QlikView charts have a straight table equivalent with the exception of the pivot table which has a more complex structure).

If the chart is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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 when each row has a unique ranking will be an integer between 1 and the number of rows in the current column segment.

In the case where several rows share the same ranking, the text and number representation 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 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 get the value corresponding to the average of the top and bottom ranking in the entire column segment.
- 1** Lowest rank on all rows.
- 2** Average rank on all rows.
- 3** Highest rank on all rows.
- 4** Lowest rank on first row, then incremented by one for each row.

The third parameter *format* specifies the text representation of the function result.

*format*

- 0** (default) Low value & ' - '& high value on all rows (e.g. 3 - 4).
- 1** Low value on all rows.
- 2** Low value on first row, blank on the following rows.

The order of rows for *mode* 4 and *format* 2 is determined by the sort order of the chart dimensions.

### Examples:

```
rank( sum( Sales ) )
rank( sum( Sales ), 2 )
rank( sum( Sales ), 0, 1 )
```

**HRank**( [ **total** ] *expression* [ , *mode* [ , *format* ] ] )

Evaluates *expression*, compares the result with the result of the other columns containing the current row segment of a pivot table and returns the ranking of the current column within the segment.

This function only works in pivot tables. In all other chart types it returns NULL.

If the pivot table is one-dimensional or if the *expression* is preceded by the **total** qualifier, 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 when each column has a unique ranking will be an integer between 1 and the number of columns in the current row segment.

In the case where several columns share the same ranking, the text and number representation can be controlled as follows:

The second parameter *mode* specifies the number representation of the function result.

*mode*

<b>0</b> (default)	If all ranks within the sharing group fall on the low side of the middle value of the entire ranking, all columns 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 columns 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 get the value corresponding to the average of the top and bottom ranking in the entire column segment.
<b>1</b>	Lowest rank on all columns in the group.
<b>2</b>	Average rank on all columns in the group.
<b>3</b>	Highest rank on all columns in the group.
<b>4</b>	Lowest rank on first column, then incremented by one for each column in the group.

The third parameter *format* specifies the text representation of the function result.

*format*

<b>0</b> (default)	Low value&' - '&high value on all columns in the group (e.g. 3 - 4).
<b>1</b>	Low value on all columns in the group.
<b>2</b>	Low value on first column, blank on the following columns in the group.

The order of columns for *mode 4* and *format 2* is determined by the sort order of the chart dimensions.

### Examples:

```
HRank( sum( Sales ) )
HRank( sum( Sales ), 2 )
HRank( sum( Sales ), 0, 1 )
```

**VRank**( [ *total* ] *expression* [ , *mode* [ , *format* ] ] )

**VRank** is actually the same function as **rank**, you may write it either way.

## Chart Field Functions

The field functions are available.

### Note!

It is not possible to use chart field functions in conjunction with *Alternate States* (page 113) as these functions do not have a *set\_expression* parameter to set the state.

**getcurrentfield** (*GroupName*)

Returns the name of the currently active field in a group.

### Example

```
getcurrentfield( myGroup )
```

**getfieldselections** ( *FieldName* [ , *ValueSep* [ , *MaxValues*] ] )

Returns a string with the current selections in a field.

*ValueSep* is the separator to be put between field values. The default is ','.

*Maxvalues* is the maximum number of field values to be individually listed. When a larger number of values is selected the format 'x of y values' will be used instead. The default is 6.

### Examples

```
getfieldselections ( Year )
getfieldselections ( Year, ';' )
getfieldselections ( Year, ';' , 10 )
```

```
getcurrentselections ([RecordSep [, TagSep [,ValueSep [, Max-
Values]]]])
```

Returns a string with the current selections in the document.

*RecordSep* is the separator to be put between field records. The default is <CR><LF>.

*TagSep* is the separator to be put between the field name tag and the field values. The default is ':'.

*ValueSep* is the separator to be put between field values. The default is ','.

*Maxvalues* is the maximum number of field values to be individually listed. When a larger number of values is selected the format 'x of y values' will be used instead. The default is 6.

### Examples

```
getcurrentselections ( )
getcurrentselections ( chr(13)&chr(10) , ' = ' )
getcurrentselections ( chr(13)&chr(10) , ' : ' , ' ; ' )
getcurrentselections ( chr(13)&chr(10) , ' : ' , ' ; ' , 10 )
```

```
getselectedcount (FieldName [, IncludeExcluded])
```

Returns the current number of selected values in a field.

If *IncludeExcluded* is true the count will include selected values, which are currently excluded by selections in other fields. If false or omitted these values will not be included.

### Examples

```
getselectedcount ( Year )
getselectedcount ( Year, true() )
```

```
getnotselectedcount ( FieldName [, IncludeExcluded])
```

Returns the current number of not-selected values in a field in a field with *AND-Mode in List Boxes (page 110)* (red values in list box).

If *IncludeExcluded* is true the count will include selected values, which are currently excluded by selections in other fields. If false or omitted these values will not be included.

### Examples

```
getnotselectedcount ( Year )
getnotselectedcount (Year,true())
```

```
getpossiblecount (FieldName)
```

Returns the current number of possible values in a field.

**Example**

```
getpossiblecount ( Year )
```

**getexcludedcount** (*FieldName*)

Returns the current number of excluded values in a field.

**Example**

```
getexcludedcount ( Year )
```

**getalternativecount** (*FieldName*)

Returns the current number of alternative values in a field.

**Example**

```
getalternativecount ( Year )
```

## Interpretation Functions

The interpretation functions interpret the contents of fields or expressions. 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.

**Note!**

For reasons of clarity, all number representations are given with decimal point as decimal separator.

The following interpretation functions are available:

<b>Num#</b>	<i>page 847</i>
<b>Money#</b>	<i>page 848</i>
<b>Date#</b>	<i>page 849</i>
<b>Time#</b>	<i>page 849</i>
<b>Timestamp#</b>	<i>page 850</i>
<b>Interval#</b>	<i>page 850</i>
<b>Text</b>	<i>page 851</i>
<b>Dual</b>	<i>page 851</i>

**Num#**

```
num# (expression [ , format-code[ , decimal-sep [ , thousands-sep ] ] )
```

The **num#** function evaluates the expression numerically according to the string given as format-code. 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 by script variables or in the operating system is used.

**Examples:**

The examples below assume the two following operating system settings:

	Default setting 1	Default setting 2
<b>Number format</b>	# ##0,#	#,##0.#

**num#( A, '#' )** where A=35,648.375 returns:

	Setting 1	Setting 2
<b>String</b>	35,648.375	35648.375
<b>Number</b>	-	35648.375

**num#( A, '#.#', '.', ',')**  where A=35,648.375 returns:

	Setting 1	Setting 2
<b>String</b>	35,648.375	35,648.375
<b>Number</b>	35648.375	35648.375

**num#( A, '#.#','!',',')**  where A=35648.375 returns:

	Setting 1	Setting 2
<b>String</b>	35648.375	35648.375
<b>Number</b>	35648375	35648375

**num#( A, 'abc#,#' )** where A=abc123,4 returns:

	Setting 1	Setting 2
<b>String</b>	abc123,4	abc123,4
<b>Number</b>	123.4	1234

## Money#

```
money#( expression [ , format-code [ , decimal-sep [ , thousands-sep ] ] ] )
```

The **money#** function evaluates the *expression* numerically according to the string given as *format-code*. 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 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 operating system settings:

	Default setting 1	Default setting 2
<b>Money format</b>	kr # ##0,00	\$ #,##0.00

**money#( , '# ##0,00 kr' )** where A=35 648,37 kr returns:

	Setting 1	Setting 2
--	-----------	-----------



<b>String</b>	35 648.37 kr	35 648.37 kr
<b>Number</b>	35648.37	3564837

**money#**( A, ' \$#', '.', ',' ) where A= \$35,648.37 returns:

	Setting 1	Setting 2
<b>String</b>	\$35,648.37	\$35,648.37
<b>Number</b>	35648.37	35648.37

## Date#

**date#**( *expression* [ , *format-code* ] )

The **date#** function evaluates the *expression* as a date according to the string given as *format-code*. 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
<b>Date format</b>	YY-MM-DD	M/D/YY

**date#**( A ) where A=8/6/97 returns:

	Setting 1	Setting 2
<b>String</b>	8/6/97	8/6/97
<b>Number</b>	-	35648

**date#**( A, 'YYYY.MM.DD' ) where A=1997.08.06 returns:

	Setting 1	Setting 2
<b>String</b>	1997.08.06	1997.08.06
<b>Number</b>	35648	35648

## Time#

**time#**( *expression* [ , *format-code* ] )

The **time#** function evaluates the *expression* as time according to the string given as *format-code*. 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
<b>Time format</b>	hh:mm:ss	hh.mm.ss

**time#**( A ) where A=09:00:00 returns:

	Setting 1	Setting 2
<b>String</b>	09:00:00	09:00:00
<b>Number</b>	0.375	-

**time#( A, 'hh.mm' )** where A=09.00 returns:

	Setting 1	Setting 2
<b>String</b>	09.00	09.00
<b>Number</b>	0.375	0.375

## Timestamp#

**timestamp#( expression [ , format-code ] )**

The **timestamp#** function evaluates the *expression* as a date and time according to the string given as format-code. 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
<b>Date format</b>	YY-MM-DD	M/D/YY
<b>Time format</b>	hh:mm:ss	hh:mm:ss

**timestamp#( A )** where A=8/6/97 09:00:00 returns:

	Setting 1	Setting 2
<b>String</b>	8/6/97 09:00:00	8/6/97 09:00:00
<b>Number</b>	-	35648.375

**timestamp#( A, 'YYYY-MM-DD hh\_mm' )** where A=8/6/97 09\_00 returns:

	Setting 1	Setting 2
<b>String</b>	1997-08-06 09_00	1997-08-06 09_00
<b>Number</b>	35648.375	35648.375

## Interval#

**interval#( expression [ , format-code ] )**

The **interval#** function evaluates the *expression* as a time interval according to the string given as a format-code. 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 value 0.99999) or smaller than 00:00:00 (numeric value 0.00000) an interval may have any value.

### Examples:

The examples below assume the following operating system settings:

Short date format: **YY-MM-DD**

Time format: **hh:mm:ss**

Number decimal separator: **.**

**interval#( A, 'D hh:mm' )** where A=1 09:00 returns:

**String** 1 09:00

**Number** 1.375

**interval#( A-B )** where A=97-08-06 09:00:00 and B=97-08-05 00:00:00 returns:

**String** 1.375

**Number** 1.375

## Text

**text** ( *expression* )

The **text** function forces the expression to be treated as text, even if a numeric interpretation is possible.

### Examples:

**text( A )** where A=1234 returns:

Result

**String** 1234

**Number** -

**text( pi() )** returns:

Result

**String** 3.1415926535898

**Number** -

## Dual

**dual** ( *s* , *x* )

Forced association of an arbitrary string representation *s* with a given number representation *x*. In QlikView, when several data items read into one field have different string representations but the same valid number representation, they will all share the first string representation encountered. This function can be used in scripts and chart expressions.

In scripting, 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.

**Note!**

If a dual value is too large to fit in a field object, it will be represented by ## and not truncated with ... like a string.

---

**Example (scripting):**

```
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.

Back to **Other Functions**.

## 69 Examples

### 69.1 Examples of Aggregate Qualifiers

These examples are made with the `sum([set_expression][distinct][total[<fld {, fld}>]] expression)` (page 791) function, but may be applied on all chart aggregation functions supporting *Set Analysis* (page 819) definitions and the **total** qualifier.

#### Example 1:

Study the representation of a straight table below, depicting the logical state before any selections have been made:

Aggregation function with total qualifier				
Month	Group	Sum(Value)	sum( total Value)	sum({1} total Value)
		<b>21</b>	<b>21</b>	<b>21</b>
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 column have the same number in all rows. This number equals the calculated total in the first expression column.

Now, let's select months 1 and 2 only. The result is as follows:

Aggregation function with total qualifier				
Month	Group	Sum(Value)	sum( total Value)	sum({1} total Value)
		<b>10</b>	<b>10</b>	<b>21</b>
1	A	1	10	21
1	B	2	10	21
2	A	3	10	21
2	B	4	10	21

The result of the third expression (fifth column) remains unchanged because the **set** definition disregards current selections. The second expression (fourth column) with the **total** qualifier, shows the new total 10, which is still equivalent to the first expression total (third column).

#### Example 2:

Study the representation of a straight table below:

total qualifier with listed fields				
Month	Group	Sum(Value)	sum( total <Month> Value)	sum( total <Group> Value)
		<b>21</b>	<b>21</b>	<b>21</b>
1	A	1	3	9
1	B	2	3	12
2	A	3	7	9
2	B	4	7	12
3	A	5	11	9
3	B	6	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 group.

## 69.2 Examples of Concat Functions

The **CONCAT()** function should not be confused with the **CONCATENATE** script keyword.

In its simplest form, **Concat()** is used to string / join together values/words/selections into one string. However, it can be utilized in a number of ways to help you solve different problems. All the examples are based upon the following table:

Table	MyColumn	Date	Value
Data	JKL	01/01/2012	11
Data1	VWX	01/02/2012	13
Data	GHI	01/03/2012	13
Data	ABC	01/04/2012	15
Data1	STU	01/05/2012	18
Data1	PQR	01/06/2012	10
Data1	MNO	01/07/2012	25
Data	DEF	01/08/2012	11

### Simple String Concatenation

As mentioned earlier, the **concat** function lets you string together a list of values. These values can be hard coded or driven by selections/data.

```
=CONCAT(MyColumn, ',')
```

```
=CONCAT(DISTINCT MyColumn, ',')
```

This simple **concat** statement would string together all of the possible values from the column **MyColumn**. You may wish to add the **DISTINCT** keyword. This would ensure that each value is only displayed once in the string.

```
ABC,DEF,GHI,JKL,MNO,PQR,STU,VWX
```

When using a simple **concat**, you have the option to add a sort weight to the function to order the string values by a column of your choice. In the example below, the date column is added to sort the values.

```
=CONCAT(MyColumn, ', ', Date)
```

```
Result: JKL,VWX,GHI,ABC,STU,PQR,MNO,DEF
```

### Concat() within an Expression/Set Statement

#### Passing multiple dynamic values to a set analysis expression

There are occasions when you want to pass a dynamic selection of values to a set statement. To do this you need to add some single quotes to the string so that the **CONCAT()** function returns e.g. 'JKL','VWX'. But you cannot have the single quotes as they are, since they then would be interpreted when the **Concat** is evaluated instead of when the set expression is evaluated. Instead, use the **Chr()** function:

```
=CONCAT(Chr(39) & MyColumn & Chr(39), ',')
```

You can then pass this **concat** statement to the inside of an expression:

```
=Sum({<MyColumn={$(=CONCAT(Chr(39) & MyColumn & Chr(39), ', '))}>} Value)
```

In most instances, this technique would be used where data islands are present. It lets you pass values to an expression that will not affect any part of the data model as the data island table is not joined to it.

### Concat() in the Script

**Concat** can also be used in the script to convert multiple rows into one single column value, just as any other aggregation.

Remembering the source data used earlier, the result of script side **CONCAT** is as follows:

```
ConcatExample:
Load Table,
Concat(MyColumn, ',') as CombinedData
Resident Temp
Group By Table;
```

The following is the result of this script function:

Table	CombinedData
Data	ABC,DEF,GHI,JKL
Data1	MNO,PQR,STU,VWX

## Using Rank() to influence the Concat()

When you start to utilize other function in conjunction with **concat()** you can start to achieve clever results. In this example, **Rank()** has been used to grab the top three performers (based on Value) and string them together.

```
=CONCAT ( IF (aggr (Rank (sum (Value)) , MyColumn) <=3 , MyColumn) , ' , ' )
```

Result: ABC,MNO,STU

## 69.3 Examples of Alternate States in Chart Expressions

These examples are intended to show best practices regarding the use of alternate states by QlikView Developers and/or extreme QlikView Power-Users. These examples can be found in the file: 'Alternate States Functionality.qvw'

### Synchronizing Selections between States

#### Example:

The following expressions can be used in a single chart:

```
count({$} DISTINCT [Invoice Number])
```

```
count({State1} DISTINCT [Invoice Number])
```

```
count({State2} DISTINCT [Invoice Number])
```

There is a problem with this method; the QlikView Developer must duplicate the selections (list boxes and multi boxes) for all three states so that the end-user can make appropriate selections for the various states. In many situations the QlikView Developer will want to have a set of 'common' selections available to all states. This will allow the end user to set the context for the various charts and then make use of specific selections to show the differences between states. Set Analysis can be used with states to keep certain selections consistent between states.

#### Example:

```
count({State1<Year = $::Year, Month = $::Month>} DISTINCT [Invoice Number])
```

```
count({State2<Year = $::Year, Month = $::Month>} DISTINCT [Invoice Number])
```

The QlikView Developer will keep the Year and Month selections in State1 and State2 synchronized with the Year and Month selections in the **default state**. The QlikView Developer can add elements to the set modifiers as necessary in order to keep even more fields consistent between states.

### Set Operators

It is possible to use set operators (+, \*, -, /) with states. The following expressions are valid and will count the distinct invoice numbers that are in either the **default state** or State1.

**Examples:**

**count**({\$ + State1} DISTINCT [Invoice Number])

counts the distinct invoice numbers in the union of the <default> state and State1.

**count**({1 - State1} DISTINCT [Invoice Number])

counts the distinct invoice numbers not in State1.

**count**({State1 \* State2} DISTINCT [Invoice Number])

counts the distinct invoice numbers in that are in both the <default> state and State1.

**Note!**

Caution should be used when using set operators in this manner. In some situations the result will not be what is expected. This is because the selections in a given state generate a set of data that may not be fully compatible with the set(s) it is being combined with. This is especially true as the complexity of the data model increases.

**Implicit Field Value Definitions**

Another way of using set operators is with the element functions P() and E(). These functions are only available within set expressions.

**Examples:**

**count**({\$<[Invoice Number] = p({State1} [Invoice Number])>} DISTINCT [Invoice Number])

This expression counts the distinct invoice numbers in the <default> state based on the possible invoice numbers available in State1.

It is almost, but not quite equivalent to the following expression:

**count**({\$<[Invoice Number] = State::[Invoice Number]>} DISTINCT [Invoice Number])

The difference between the expressions is that in the first one, the possible values on invoice number from State1 are passed into the **default state**. In the second expression, the selected values from State1 are passed into the **default state**. If the user has not made any invoice number selections in State1, no invoice number values will be passed into the **default state**.

Set operators are best used within set modifiers. The following expression finds the intersection of the possible invoice numbers from State1 and State2 and passes those values into the invoice numbers in the **default state**.

**Examples:**

**count**({\$<[Invoice Number] = p({State1} [Invoice Number]) \* p({State2} [Invoice Number])>} DISTINCT [Invoice Number])

The expression to find the intersection of invoice numbers in the <default> state and State1 is:

**count**({\$<[Invoice Number] = p({\$} [Invoice Number]) \* p({State1} [Invoice Number])>} DISTINCT [Invoice Number])

This expression may seem confusing because it uses a state (<default> in this instance) in both the element function and a set identifier. Keep in mind that the element function p(\$\$) is returning the possible values in the **default state**. The set/state identifier {\$\$} is being modified by the result of the element functions. Any invoice selections currently existing in the **default state** are being overridden by the values from the intersection of the element functions.

It should be noted that the above expression is still not completely correct as it fails to synchronize the common selections between the **default state** and State1. Below is an expression that will do so:

**Examples:**

**count**({\$<[Invoice Number] = p({\$} [Invoice Number]) \* p({State1<Year = \$::Year, Month = \$::Month>} [Invoice Number])>} DISTINCT [Invoice Number])



As before, the QlikView Developer can add fields into the modifier to keep selections consistent between various states.

## 69.4 Examples of Chart Ranking Functions

These examples are made with the **rank (VRank)** function, but may be applied in a similar manner to the **HRank** function. However, note that the **HRank** function is relevant only to pivot tables.

### Example 1:

Study the two one-dimensional straight tables below:

Single dimension RANK		
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

Sorted by ranking column		
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 first one is sorted by the first column while the second one is sorted by the last column. This exemplifies the basic functionality of **rank**. The highest value has the highest rank (lowest ranking number).

The **rank** function always returns 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. It is now possible to see how the current column segment is the inner grouping in the multi-dimensional case. The Month field within group A is ranked separately from the Month field 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 column 4 - 8 shows the number representation of the same ranking in different modes. The expression in each column is:

`num( rank( sum( X ), 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 is therefore rounded downwards to 2. Rows 4 and 5 share ranking too, but fall just above the middle of the ranking table. Therefore they get a number representation of the average of the first and the 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 data ranking highest and lowest within a group.

**Mode 1** In both cases the lower ranking figure within the group is used, i.e. 2 for rows 2 and 3, 4 for rows 4 and 5.

**Mode 2** In both cases the average of the low and high ranking within the group is used, i.e. 2.5  $((2+3)/2)$  for rows 2 and 3, 4.5  $((4+5)/2)$  for rows 4 and 5.

**Mode 3** In both cases the higher ranking figure within the group is used, i.e. 3 for rows 2 and 3, 5 for rows 4 and 5.

**Mode 4** Each row is assigned its own distinct numeric value. The order within 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:

The effect of the format parameter on the result text representation				
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 chart's dimensions.

## 69.5 Examples of Chart Inter Record Functions

### Top Function Examples

These examples are made with the **top** function, but may be applied in a similar manner to the **bottom**, **first** and **last** functions. However, note that the **first** and **last** functions are relevant only to pivot tables.

#### Example 1:

Study the straight table below, depicting the use of the **top** function in a one-dimensional table:

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 included.)

Note that expressions using the **top** function will be properly evaluated in the total row too since the total has a clear relation to a specific column segment, in this case the entire column.

### Example 2:

Below is a two-dimensional straight table sorted primarily on the field 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 the multi dimension case, you may again refer to the absolute top row of the table with the same value being returned for all rows. The expression will of course be evaluated for the column segment spanning the entire column.

The expression using the **top** function without the **total** qualifier will evaluate to NULL in the total row, because it cannot be clearly associated with a specific column segment.

We will now convert the above straight table to a pivot table with all totals activated.

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 total row, because 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 **total** qualifier will lack values in the partial totals but will return a value in the grand total row.

### Example 3:

Study the following straight table sorted on the field 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%

We may continue by changing the inter field sort order so that the chart is sorted primarily on the field Month. The table will now look like this:

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 Examples

These examples are made with the **above** function, but may be applied in a similar manner to the **below**, **before** and **after** functions. However, note that the **before** and **after** functions are relevant only to pivot tables.

### Example 1:

Study the straight table below, depicting the use of the **above** function in a one-dimensional table:

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. The **above** function returns NULL on the first row, 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:

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 (third column) will only act within each sort group. A NULL value will be returned on the top row of each column segment.

When a **total** qualifier is added (fourth column), the entire column will be regarded as one column segment. Only the top row will return NULL. All total rows are disregarded and return NULL.

## RowNo and NoOfRows Function Examples

This example is made with the **RowNo** and **NoOfRows** functions, but may be applied in a similar manner to the **ColumnNo** and **NoOfColumns** functions. However, note that the **ColumnNo** and **NoOfColumns** functions are relevant only to pivot tables.

### Example 1:

Study the two-dimensional pivot table below:

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 will return the row number within each sort group column segment. In subtotal rows, the row number 0 will be returned, because these totals clearly belong to a specific column segment. NULL will be returned in the grand total row.

**Column 4** With the total qualifier, the **RowNo** function will return the row number within the entire column. In subtotal rows a NULL value will be returned. In the grand total row 0 will be returned.

**Column 5** The **NoOfRows** function will return the number of data rows within each sort group column segment. In subtotal rows, the same number will be returned as in data rows. NULL will be returned in the grand total row.

**Column 6** With the total qualifier, the **NoOfRows** function will return the number of data rows within the entire column, which is the same as what will be returned in the grand total row. In subtotal rows NULL will be returned.

### **dimensionality ( )**

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.

1 for 2nd level partial sums and entries only expanded in the 1st column.

0 for grand total cell.

### **dimensionality ( )**

Pivot table			
Country	Team	Salesman	dimensionality()
Denmark	Copenhagen	Preben	3
		Total	2
	Total		
Germany			1
Sweden	Gothenburg	John	3
		Total	2
	Lund		2
	Total		
Total			0

Straight table			
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 ( )**

Pivot table				
Country		Denmark	Germany	
Salesman		Total	Preben	
Team			Total	Copenhagen
	0	1	2	3
				1



# 70 Nested Aggregations and Related Issues

This chapter exemplifies some important techniques in relation to nested aggregations and the use of the *Advanced Aggregation (page 301)* function in charts.

---

**Note!**

As of QlikView version 9 no more than 100 levels of nesting is allowed.

---

## 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 an 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. This type of nesting need is quite common and should be used wherever suitable.

## Nested Aggregations with the Aggregation Function

Nesting with **total** is not always enough. For more generic nesting capabilities you will have to use the *Advanced Aggregation (page 301)* function in combination with calculated dimensions, see *Add calculated dimension... (page 622)*.

**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

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

Now however, let's ask a couple of new questions based on the knowledge just gained: "How may sales reps have only a 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 the *Advanced Aggregation (page 301)* 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? -	
=aggr(count(Customer),SalesRep)	count(distinct SalesRep)
1	2
2	3
3	1
5	1

Two things are 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 *Advanced Aggregation* (page 301) function may of course contain calculated dimensions, which in turn make use of the *Advanced Aggregation* (page 301) function. It should however be relatively easy to loose track of what you are doing when passing the third level of aggregation.

## 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.

Having issued that warning, we will now look at an example where the sum of rows total is nevertheless the desired result.

### Example:

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 the 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

We must now make a chart with Class as dimension and **max**(Score) as expression. A straight table with sum of rows will look like follows:

Straight table with sum of rows	
Class	max(Score)
History	3
Math	4
English	5
	12

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 as follows:

Pivot table with expression total	
Class	max(Score)
English	5
History	3
Math	4
Total	5

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 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 like this:

Pivot table with sum of rows	
Class	sum(aggr(max(Score),Class))
English	5
History	3
Math	4
Total	12

As you see the total is again the one desired. What happened?

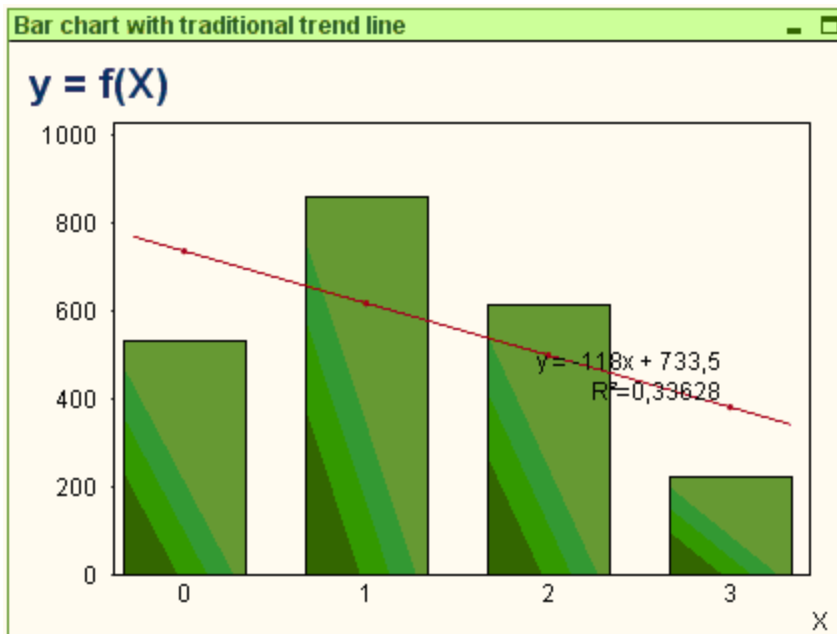
Well the beauty of the *Advanced Aggregation (page 301)* function is that in the individual rows it will evaluate to only a single value. This is because the dimension obviously only has one possible value on each ordinary data row. Since the inner dimension and expression are the same as for the surrounding chart, each value will of course be exactly the same as the result without the enclosing **sum** and *Advanced Aggregation (page 301)* functions.

For the total row however, the *Advanced Aggregation (page 301)* 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.

## 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 **Chart Properties**. It is also possible to display the regression equation.

### Example:



If you want to display the regression data in e.g. a table chart, the regression must be calculated. The `linest_m` (`{{set_expression}} [distinct] [total [<fld {fld}>] ] y-expression, x-expression [, y0 [, x0 ]]`) (page 796) and `linest_b` (`{{set_expression}} [distinct] [total [<fld {fld}>] ] y-expression, x-expression [, y0 [, x0 ]]`) (page 796) 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 *Advanced Aggregation* (page 301) function containing the same base expression and dimension(s) as the containing chart. The *Advanced Aggregation* (page 301) 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 where the regression is shown as a regular line expression.



Note that the trend line here is not a traditional QlikView trend line, but a regular expression plotted as 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 to a straight table, where the regression values are shown in cells.

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

In the straight table above three extra columns have been added to show the  $m$ ,  $b$  and  $R^2$  values. These of course are constant for all table rows. The expressions needed would look 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)
```

# 71 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.

## 71.1 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 *Edit Expression Dialog (page 757)*).

## 71.2 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”.

## 71.3 Calculated Formula

The syntax for *expression* in calculated labels is virtually the same as for *Chart Expressions (page 785)*, with a few exceptions:

- With no dimensions to iterate over, the *Chart Aggregation functions (page 790)* will essentially behave as if used in chart expressions with the **total** qualifier in front of all field names. The **total** qualifier is thus optional and has no special meaning in calculated formulas. The **set** definitions work just as in chart expressions, i.e. they cause aggregations over a selection different than the current one.
- In a calculated formula, field names may optionally be used without an enclosing aggregation function. In this case **only** will be used as an aggregation function.

### Example:

```
= Currency
```

is equal to

```
= only(Currency)
```






## 72 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 *Chart Properties: Dimensions (page 620)*).

Any fields can be grouped together.

Drill-down groups are marked with an  icon,

while cyclic groups are displayed with an  icon.

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.

### 72.1 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 are:

*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 *Fields (page 171)* 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 field becomes available 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.

A special feature appears when the same drill-down group is used in more than one dimension in the chart specification. The second time the group occurs, the field used will automatically be taken from one step down the group's field list. For example, if you create a two-dimensional chart with the group *Geography* above used as both Main dimension and 2nd dimension, *Continent* and *Country* will initially be used. As soon as one single continent is selected, *Country* and *State* will be used instead.

---

**Note!**

Although it is convenient to use this feature with natural hierarchies, there is nothing preventing its use in other situations as well.

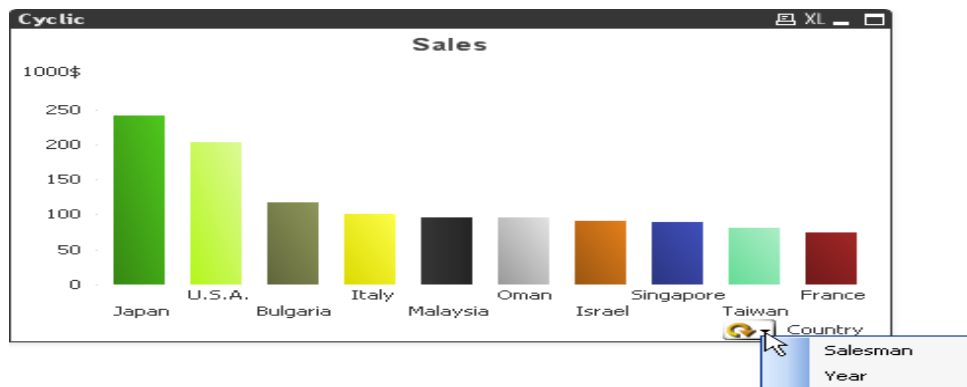
---

**Note!**

If you use a dimension expression that results in a data island it is not possible to drill down, as a valid hierarchy is required. In this case you need to adapt the expression to incorporate the values in the hierarchy.

---

## 72.2 Non-Hierarchic Groups (Cyclic)

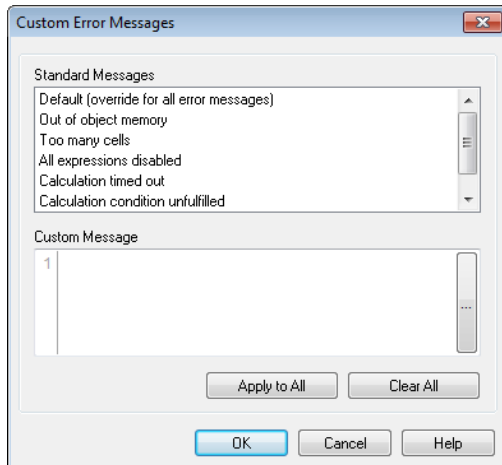


Sometimes you may find it useful to group fields which do not form a natural hierarchy. The reason would be to enable the user to make quick changes to the data to be displayed in the chart or list box.

Any fields can be grouped together in a cyclic group. When a cyclic group is used as a chart dimension, the chart will initially use the first field in the list. The user may then switch to another field by clicking the cycle icon in the chart. The fields are displayed in the order in which 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 again. In this way the chart may cycle through the fields indefinitely.

Do not confuse cyclic groups with **Cyclic Group** in chart expressions (see: *Chart Properties: Expressions (page 629)*)! Cyclic dimension groups are made up of a number of *Fields (page 171)*, whereas the cyclic display in chart expressions cycles through a number of *Chart Expressions (page 785)*.

## 73 Custom Error Messages



*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.

- 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 Messages**      Here you enter the text to be shown instead of the standard message selected above. The text may be a calculated formula.  
This feature makes it possible to e.g. change a rather cryptic error message (like: *Local heap space exhausted*) into a helpful hint for tackling the problem (in this case: *Please make a selection...* or: *Please increase the allocation of memory / cells by...* ).
  
- Apply to All**              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.



---

# Part 7 Number Formats

## 74 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.

### 74.1 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.

### 74.2 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.

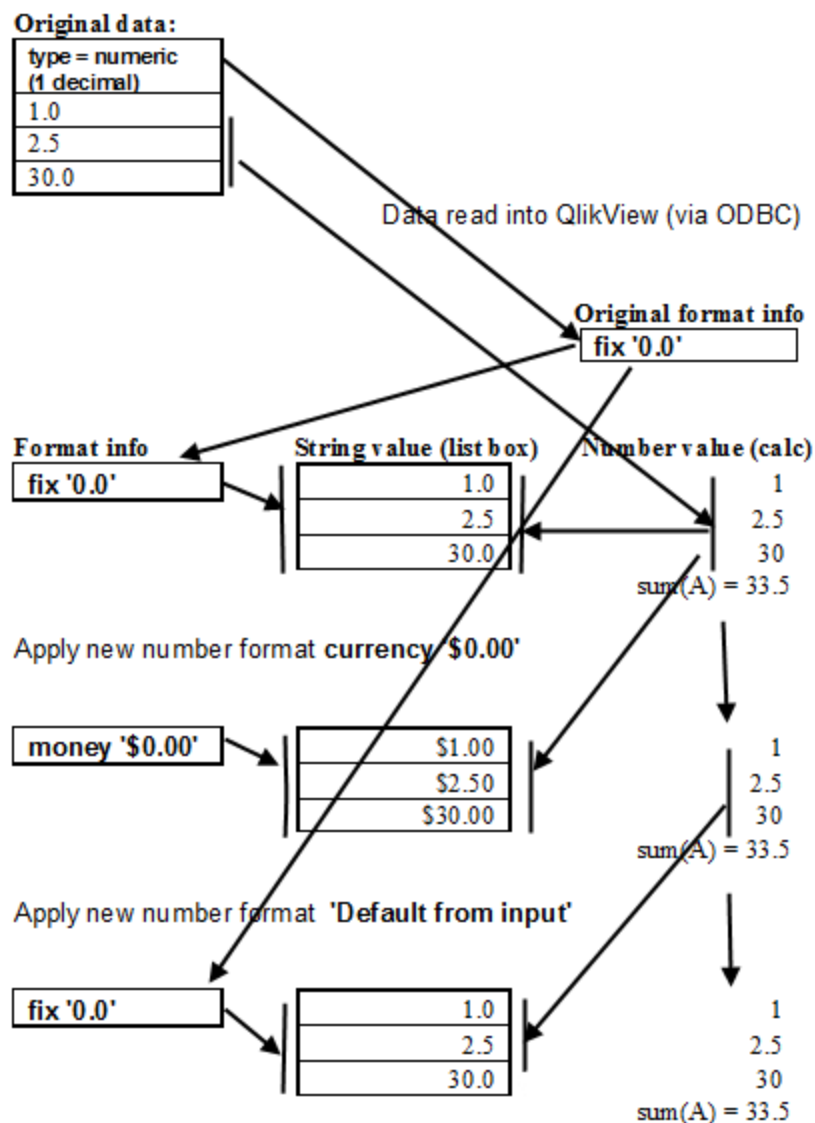
QlikView will remember the original number format of the field even if the number format is changed in the number format dialogs of the application. The original format can always be restored by clicking on the **Default from Input** button in the number format dialogs.

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 (*Number Interpretation Variables (page 366)*) or the operating system settings (Control Panel).

#### Example:



## 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 *Number Interpretation Variables (page 366)*.)

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 *Number Interpretation Variables (page 366)*) are correctly defined and that the system settings in the Control Panel are correct.

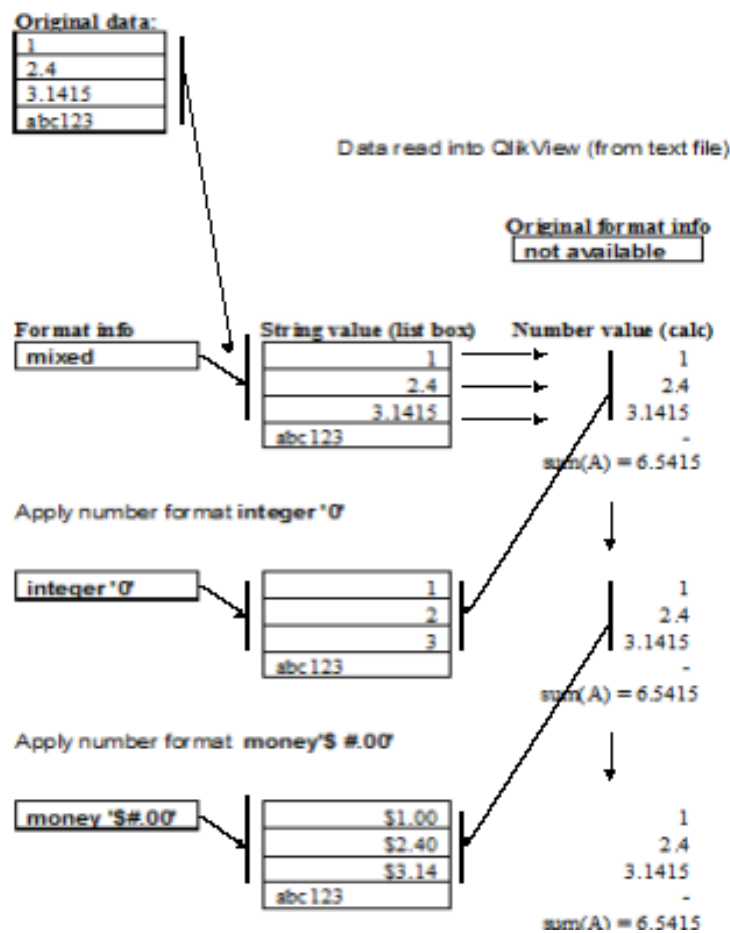
Once QlikView has interpreted data as a date or time, it is possible to change to another date or time format in the properties dialog for the sheet object.

Since there is no predefined format for the data, different records may of course contain differently formatted data in the same field. It is possible e.g. to 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:**





## 74.3 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 calculations 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 *Number Interpretation Variables (page 366)*) or with the help of a formatting function, see *Formatting Functions (page 370)*. 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)**

0.375
33857
97-08-06
970806
08/06/1997

**Number value**

-
-
-
-
35648

B4) With number format time 'hh:mm'

**String value (list box)**

0.375
33857
97-08-06
970806
00:00

**Number value**

-
-
-
-
35648

B5) With number format fix '# ##0.00'

**String value (list box)**

0.375
33857
97-08-06
970806
35 648.00

**Number value**

-
-
-
-
35648

# 75 Format Codes for Interpretation and Formatting Functions

In number format controls 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.

## 75.1 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 thousands separator or the decimal separator, use the thousands separator and the decimal separator.

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, for example, a format string of "0000-0000-0000" (thousand separator="-") could be used to display a twelve-digit part number as "0012-4567-8912".

## 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.

## 75.2 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.

## 75.3 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 "t" after the time.
- Arbitrary separators can be used.

## 75.4 Time Stamps

The same notation as that of dates and times above is used.

## Examples of Format Codes

### Examples (Numbers):

# ##0 describes the number as an integer with a thousands separator.

###0 describes the number as an integer without a thousands 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.

### Examples (Special Number Formats):

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	excix
num(199, '(ROM)')	returns	CXCIX

**Examples (Date):**

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/YY 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.

WWWY YY-MM-DD describes the date as Saturday 01-03-31.

**Examples (Time):**

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

**Examples (Timestamps):**

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 8 Macros and Automation

# 76 Using Automation and Macros with QlikView

## 76.1 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.

## 76.2 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.



## 77 Internal Macro Interpreter

QlikView objects are accessible via automation from inside QlikView by means of the built-in scripting engine.

### 77.1 Invoking Macros

Macros written in VBScript or JScript, inside a QlikView document can be invoked in a number of different ways:

#### Document Events:

- A macro can be run after opening a QlikView document.
- A macro can be run after script re-execution.
- A macro can be run after the **Reduce Data** command.
- A macro can be run after a selection in any field in a document.
- A macro can be run when the value of any variable changes in a document.

Macros that are invoked from document events are created from the *Document Properties: Triggers (page 440)* page.

#### Sheet Events:

- A macro can be run after a sheet is activated.
- A macro can be run when a sheet is deactivated.

Macros that are invoked from sheet events are created from the *Sheet Properties: Triggers (page 468)* page.

#### Sheet Object Events:

- A macro can be run after a sheet object is activated.
- A macro can be run when a sheet object is deactivated.

Macros that are invoked from sheet object events are created from the *Sheet Properties: Triggers (page 468)* page.

#### Button Events:

- A button can be defined as a macro button.

A macro that is invoked from a macro button event is created from the button's *Actions (page 552)* page.

#### Field Events:

- A macro can be run when a selection has been made in a specified field, see *Fields (page 171)*.
- A macro can be run when a selection is made in any field which is logically associated to a specified field.
- A macro can be run when selections are locked in a specific field.
- A macro can be run when selections are unlocked in a specific field.

Macros that are invoked from field events are created from the *Document Properties: Triggers (page 440)* page.

## Variable Events:

- A macro can be run when a new value is directly entered into a specified *Variable (page 173)*.
- A macro can be run when the value of the specified variable changes as a result of changes in other variables or the logical state of the document.

Macros that are invoked from variable events are created from the *Document Properties: Triggers (page 440)* page.

## Macro Editor:

Irrespective of how they are triggered, macros are created and edited in the *Edit Module (page 895)* dialog.

## 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
```

## 77.2 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 you 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

```

## 77.3 Edit Module

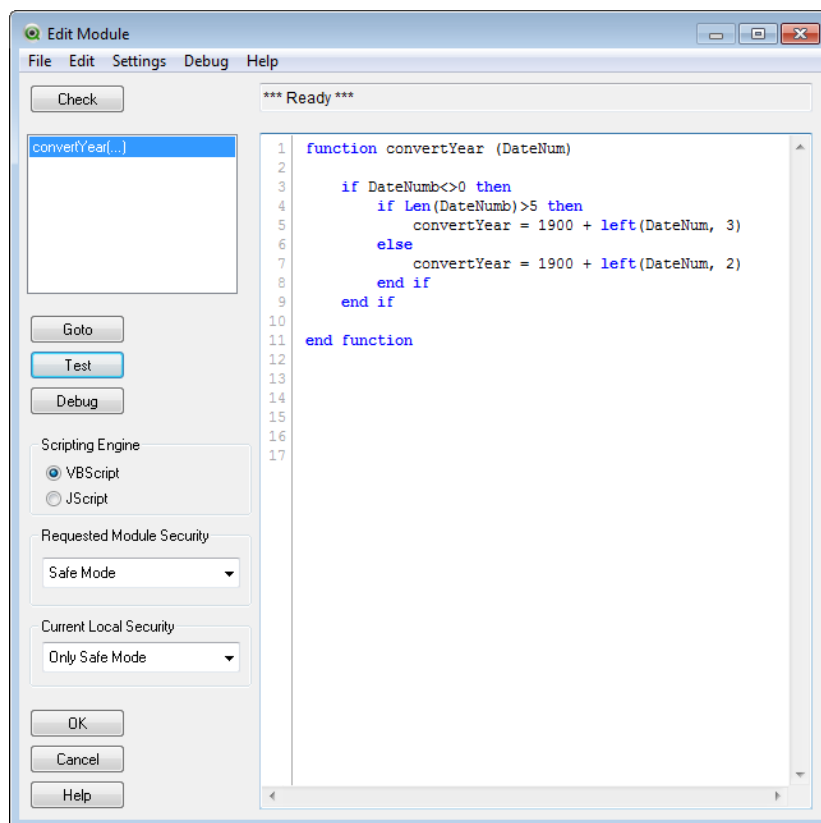
Macros and custom defined functions can be written in VBScript or JScript using the **Edit Module** dialog. The module is saved with the document.

The following commands are available:

From the **File** menu the **Export to Module File** command lets you save the script as a file. There is also an option to **Print** the script.

The **Edit** menu contains a number of common options: **Undo**, **Redo**, **Copy**, **Cut**, **Paste**, **Clear**, **Select All** and **Find/Replace**. There is also an option: **Insert File** which permits you to import a previously saved Script File. Transform any text row in the script into a comment by means of the **Comment** command and switch it back again by using **Uncomment**.

The *User Preferences: Editor* (page 74) dialog is opened by choosing the **Configure** command from the **Settings** menu.



*The Edit Module dialog*

### (Macro Edit Box)

The large edit box is where you type your macros. All macros should be written as sub-routines between a matching pair of `sub .. end sub`.

<b>(List of entry points)</b>	As soon as an entry point is recognized by the scripting engine, it will appear in the list in the smaller, left text box. It is also possible to select entry points from this list.
<b>(Message)</b>	The current state of any error messages is displayed in the box above the macro edit box.
<b>Check</b>	Once you have a written sub, check and validate its syntax by pressing this button.
<b>Goto</b>	Highlighting an entry point from the <b>List of entry points</b> and clicking <b>Goto</b> , takes you directly to its position in the <b>Macro Edit Box</b> .
<b>Test</b>	With an entry point selected, you may test a macro by clicking the <b>Test</b> button. Any errors will appear in the <b>Message</b> box.
<b>Debug</b>	Begin by selecting one of the subroutines to be debugged. By clicking the <b>Debug</b> button, the debug mode is entered.
<b>Scripting Engine</b>	Select between <b>VBScript</b> and <b>JScript</b> .
<b>Requested Module Security</b>	By selecting <b>Safe Mode</b> the document designer indicates that the macros in this QlikView document contain no code that can access either the system or applications outside QlikView. If such code is encountered during macro execution in a document declared to be in <b>Safe Mode</b> , the execution will fail. By selecting <b>System Access</b> mode the document designer ensures that the end user will be prompted when opening the document to either approve system access ( <b>Allow System Access</b> ), disable all macros in the document ( <b>Disable Macros</b> ) or allow only macros without system access ( <b>Safe Mode</b> ).
<b>Current Local Security</b>	When opening a document declared to be in safe mode but containing potentially "unsafe" macro code, the user will be prompted (see <i>Requested Module Security</i> above). The user's action will be remembered by the system but this setting can be changed at a later time from this menu. Macro security level can be set to <b>Don't Run at All</b> , <b>Only Safe Mode</b> or <b>Allow System Access</b> .

---

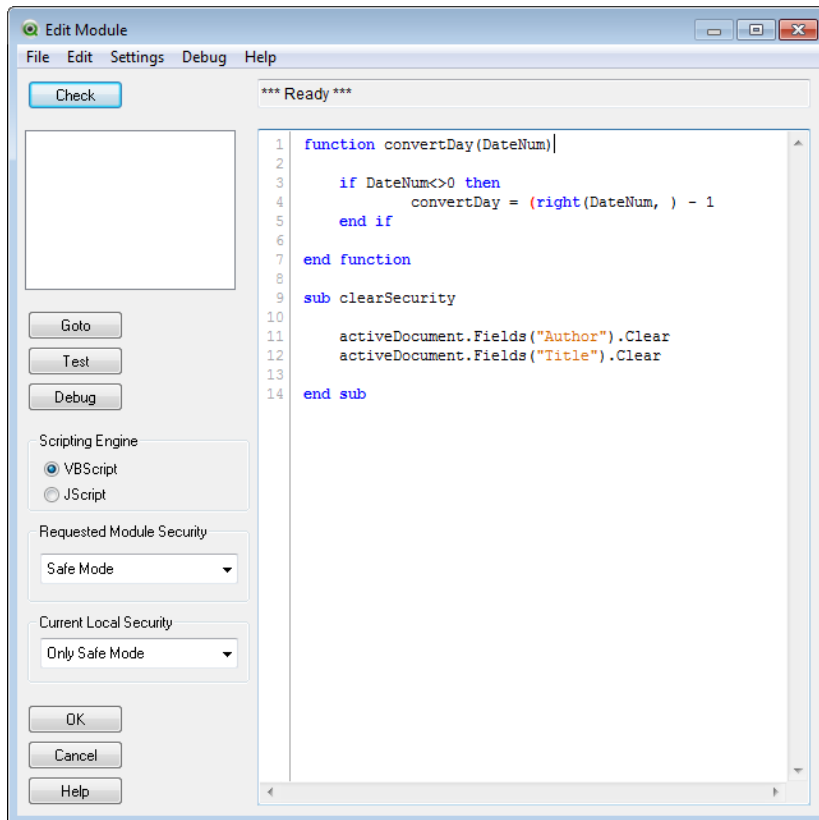
**Note!**

This is just a very brief introduction to the QlikView Scripting Engine. For a deeper understanding of VBScript or JScript you should refer to a handbook on the subject.

---



## 77.4 Macro Debugger



*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.

- |                  |  |
|------------------|--|
| <b>Continue</b>  | Continues execution of the macro to the next breakpoint or the end of the macro. Equivalent to pressing F5.  |
| <b>Step Into</b> | 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. |

<b>Step Out</b>	Executes the remaining statements of the current subroutine. Equivalent to Ctrl+Shift+F8.
<b>Step Next</b>	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.
<b>Variable Window</b>	Opens a window where the values of different variables can be inspected and changed.
<b>Expression Window</b>	Opens a dialog where you may enter and evaluate a VBScript or JScript expression based on current variable values.
<b>Clear Breakpoints</b>	Clears the breakpoints set in the code for debugging. Equivalent to pressing Ctrl+Shift+F9.
<b>Stop Debugging</b>	Exits debug mode and returns to the normal Edit Module dialog.

## 78 Getting Hold of a QlikView Document

### 78.1 Accessing QlikView Documents from the Outside

QlikView documents can be accessed from outside the QlikView program by means of the Automation interface.

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
```

### 78.2 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 Clr  
Set QVDoc = ActiveDocument  
QvDoc.ClearAll(false)  
End sub  
Sub EndQV  
ActiveDocument.Application.Quit()  
End sub
```



## 79 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 function 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);
```

## 79.1 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.

# 80 Using Macros in QV Documents on the QV-Server

## 80.1 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 QlikView as a client all macros are executed in the client. In this case more operations can be permitted.

---

**Note!**

Using macros extensively may lead to potential problems with macros executing in parallel instead of as sequentially, as a designer might have intended/assumed.

---

## 80.2 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 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
- 

**Note!**

Layout operations acting on the properties of Server objects are supported.

---

## 80.3 Macro trigger limitations

The following triggers will work as usual in the QlikView Server environment for all clients, except AJAX mode which doesn't support event triggers:

```
Document.OnAnySelect  
Field.OnSelect  
Field.OnChange  
Field.OnUnlock  
Document.OnPressMacroButton  
Variable.OnChange  
Variable.OnInput
```

The following triggers are not to be used in the QlikView Server environment, since they lack meaning or may cause unexpected results:

```
OnPostReduceData  
OnPostReload
```

---

## 80.4 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.

## 80.5 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.



---

# Part 9 Appendix

## 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 is one inherent limitation in QlikView that you must be aware of when building very large documents: A QlikView document cannot have more than 2,147,483,648 distinct values in one field.

The number of fields and tables as well as the number of table cells and table rows that can be loaded, is limited only by RAM.

When running the 32-bit version of QlikView, RAM is limited to 2, 3 or 4 GB, depending on operating system version and configuration, regardless of the size of physical memory in the computer. This limitation does not exist when running the 64-bit version of QlikView.

### A.2 System Requirements

QlikView is designed to run on Windows™ operating systems.

You can read more about the hardware and software requirements to run QlikView on your computer on <http://www.qlikview.com/download>.









### A.3 Dynamic Link Libraries (DLLs) Needed

In the Windows system directories of all the clients there must be a number of DLLs. 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 DLLs 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

<b>Ctrl+N</b>	Equivalent to <b>New</b> on the <b>File</b> menu.	
<b>Ctrl+O</b>	Equivalent to <b>Open...</b> on the <b>File</b> menu.	
<b>Ctrl+Shift+O</b>	Equivalent to <b>Open in Server...</b> on the <b>File</b> menu.	
<b>Ctrl+S</b>	Equivalent to <b>Save</b> on the <b>File</b> menu.	
<b>F12</b>	Equivalent to <b>Save As...</b> on the <b>File</b> menu.	
<b>Ctrl+P</b>	Equivalent to <b>Print...</b> on the <b>File</b> menu.	
<b>Ctrl+Shift+P</b>	Equivalent to <b>Print As PDF...</b> on the <b>File</b> menu.	
<b>Ctrl+E</b>	Equivalent to <b>Edit Script...</b> on the <b>File</b> menu.	
<b>Ctrl+R</b>	Equivalent to <b>Reload</b> on the <b>File</b> menu.	
<b>Ctrl+Shift+R</b>	Equivalent to <b>Partial Reload</b> on the <b>File</b> menu.	
<b>Ctrl+T</b>	Equivalent to <b>Table Viewer...</b> on the <b>File</b> menu.	

### B.2 Edit Menu Command Shortcuts

<b>Ctrl+Z</b>	Equivalent to <b>Undo Layout Change</b> on the <b>Edit</b> menu.	
<b>Ctrl+Y</b>	Equivalent to <b>Redo Layout Change</b> on the <b>Edit</b> menu.	
<b>Ctrl+X</b>	Equivalent to <b>Cut</b> on the <b>Edit</b> menu.	
<b>Ctrl+C</b>	Equivalent to <b>Copy</b> on the <b>Edit</b> menu.	
<b>Ctrl+V</b>	Equivalent to <b>Paste</b> on the <b>Edit</b> menu.	
<b>Del</b>	Equivalent to <b>Remove</b> on the <b>Edit</b> menu.	
<b>Ctrl+A</b>	Equivalent to <b>Activate All</b> on the <b>Edit</b> menu.	
<b>Ctrl+F</b>	Equivalent to <b>Search</b> on the <b>Edit</b> menu.	
<b>Ctrl+Shift+F</b>	Equivalent to <b>Advanced Search</b> on the <b>Edit</b> 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+Left arrow** Equivalent to **Back** on the **Selections** menu.

**Shift+Right arrow** 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)

<b>Ctrl+L</b>	Equivalent to <b>Lock</b> on the <b>Object</b> menu (locks selections in active object).
<b>Ctrl+U</b>	Equivalent to <b>Unlock</b> on the <b>Object</b> menu (unlocks selections in active object).
<b>Ctrl+D</b>	Equivalent to <b>Clear</b> on the <b>Object</b> menu (clears selections in active object).
<b>Alt+Enter</b>	Equivalent to <b>Properties...</b> on the <b>Object</b> menu (opens the <b>Properties</b> dialog of an 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.

<b>Ctrl+G</b>	Go to line number in the script.
<b>Ctrl+K,C</b>	Comment lines in script.
<b>Ctrl+K,U</b>	Uncomment lines in script.
<b>Ctrl+Q,T,A</b>	Add tab in script.
<b>Ctrl+Q,T,P</b>	Promote active tab.
<b>Ctrl+Q,T,D</b>	Demote active tab.
<b>Ctrl+Q,T,N</b>	Rename active tab.
<b>Ctrl+Q,T,R</b>	Remove active tab.
<b>Ctrl+Q,Q</b>	Creates an autogenerated script.
<b>Ctrl+Q,U,I</b>	Opens Unicode input utility.
<b>Ctrl+Q,J,P,G</b>	Creates a script for reading attributes from jpeg files.
<b>Ctrl+Q,M,P,3</b>	Creates a script for reading attributes from mp3 files.
<b>Ctrl+Q,W,M,A</b>	Creates a script for reading attributes from wma files.

## B.10 F Key Keyboard Shortcuts

<b>F1</b>	Activates context sensitive help.
<b>F3</b>	Enters search mode if a searchable object is activated.
<b>F6</b>	Activates the immediate left sheet tab of the currently active tab.
<b>F7</b>	Activates the immediate right sheet tab of the currently active tab.
<b>F12</b>	Equivalent to <b>Save As...</b> on the <b>File</b> menu.
<b>Ctrl+F6</b>	Activates the leftmost sheet tab.
<b>Ctrl+F7</b>	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 in terms of performance. As the amount of data grows, both time and memory constraints may become very evident if the application is poorly designed. It is possible that some simple design alterations improve performance substantially. 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 the 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, and 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 *Using the Extended IntervalMatch Syntax to Resolve Slowly Changing Dimension Problems* (page 395)).

## 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.

QlikView sorts strings of mixed characters and numbers in alphanumeric order. That is, numbers are sorted in value order while non-numbers are sorted in ASCII order, as opposed to traditional ASCII-only sort order.

Example:

ASCII sort	Alphanumeric sort
A1	A1
A10	A4
A11	A5
A30	A6
A4	A10
A5	A11
A6	A30

## 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 3.1 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 a software component used for software installation, modification, repair and removal on Microsoft Windows systems. 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" or equivalent tab.

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](http://msdn.microsoft.com)

[www.itninja.com](http://www.itninja.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 QlikView documents created in old QlikView versions in later versions of QlikView?

A: Yes, you can read all files created in version 7.52 and later. If you need to convert files from earlier versions of QlikView, you need QlikView 7.52. Contact your program vendor for a free copy. QlikView 11 shares file format with QlikView 7, 8, 9 and 10.

Q: Can I run QlikView documents created in QlikView 11 in earlier versions of QlikView?

A: Usually yes! QlikView 11 files can be used directly by QlikView 10, 9, 8 and 7. QlikView 11 can also save files in QlikView 10, 9, 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. See *Renaming Fields (page 384)*.

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 internal 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 internal 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. See *File Wizard: Type (page 198)* and *File Wizard: Type - Fixed Record (page 200)*.

Q: Can I update a QlikView document by loading only the data that have changed?

A: Yes, see *QVD Files (page 409)* for instructions on how to make incremental reloads.

Q: Can I read tables from web pages into QlikView?

A: Yes, the table file wizard (see *File Wizard: Type - HTML (page 202)*) 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 internal table. It is therefore impossible for QlikView to know how to calculate frequency and other statistical entities for the field. It is also possible to 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 internal 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 internal 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 enable the AND mode, the field must:

- exist in only one internal table,
- be the second field of only two fields in that table and
- contain no duplicate records.
- it must be preceded 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. 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 in QlikView. 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. See *Circular References (page 111)* for further information.

## D.5 Layout

Q: Why cannot sheet objects without caption be sized at its upper end?

A: When you turn off the caption for a sheet object the upper border can only be used for moving the sheet object around, so instead use the upper corners of a sheet object to size.

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. 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 sheet 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 QlikView 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" (see *Section (page 269)* and *Access Restriction Table Wizard (page 195)*) 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. See *Security (page 417)*.

Q: Where do I store information on access rights?

A: Information can be stored as a text file in a protected unit, or as a table in a database. It is also possible to store it as an inline statement in the script (see *Inline Data Wizard (page 194)*).

Q: Is there an automated way to create personalized copies to a large group of users?

A: Yes, you need a QlikView Publisher which is an add-on module to QlikView Server.



# 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.

## E Intentionally Creating Loosely Coupled Tables

In some special situations, you might want to disconnect some data from the normal QlikView logic. You can use loosely coupled tables to restrict selections in one field to propagate through to the other fields in the table.

To set a table to be loosely coupled, open the **Document Properties** dialog and select the **Tables** tab.

This chapter includes some examples of how loosely coupled tables alter the QlikView logic.

Each of the following three table boxes represents a table read into QlikView:

Table1		Table2		Table3	
B	A	A	C	C	D
1	x	x	6	6	a
2	y	y	7	7	b
3	z	z	8	8	c

If the value 2 is selected in field B the following will happen:

Table1		Table2		Table3	
B	A	A	C	C	D
2	y	y	7	7	b

The selection ripples through all tables, and excludes values that are not associated.

Now, let's keep this selection but make Table2 loosely coupled. This means that the logic will be cut between the fields A and C in Table2. The result will look this:

Table1		Table2		Table3	
B	A	A	C	C	D
2	y	y	6	6	a
		y	7	7	b
		y	8	8	c

Note that Table2 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.

### Example:

The three tables below are found in a rather typical structure: one transaction table and two dimension tables associating to it via one field each.

You can visualize sales per year and product group using a pivot table. Next to the pivot table, two list boxes show the dimension fields:

sum(Amount)			Year	ProdGrp
Year	ProdGrp	sum(Amount)	2011	X
2011	X	36	2012	Y
	Z	14		Z
		<b>50</b>		
2012	X	45		
	Y	13		
		<b>58</b>		
<b>Total</b>		<b>108</b>		

Even though this is a correct pivot table, the effects of QlikView's logic could now potentially lead to undesired results. If the year 2012 is selected, you get the following result:

sum(Amount)			Year	ProdGrp
Year	ProdGrp	sum(Amount)	2011	X
2012	X	45	2012	Y
	Y	13		Z
		<b>58</b>		
<b>Total</b>		<b>58</b>		

Product group Z is no longer visible in pivot table. This is natural, since the value Z in the field ProdGrp has been excluded by the selection of the value 2012 in the field Year. However, when you analyze sales of year 2012, you probably want to see 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 2012.

In some sense, the two fields Year and ProdGrp have nothing to do with each other, and therefore should not interact just because they happen to be associated via the Trans table. This can be dealt with by declaring the Trans table as loosely coupled, which changes the layout of the pivot table:

sum(Amount)			Year	ProdGrp
Year	ProdGrp	sum(Amount)	2011	X
	X	45	2012	Y
2012	Y	13		Z
	Z	0		
	<b>Total</b>	<b>58</b>		
<b>Total</b>		<b>58</b>		

Note that the selection in the Year list box does not render any value in the ProdGrp list box anymore.

Note:

Make sure that **Suppress Zero-Values** is not selected for dimensions on the **Presentation** tab of the **Chart Properties** dialog.

## See also:

*Circular References (page 111)*

# F Backus-Naur Formalism

The QlikView command line syntax ( *Command Line Syntax (page 36)*) and *Script Syntax (page 219)* 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};
```





---

# Glossary

## A

---

### **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.

### **access restriction**

Method to restrict access of a QlikView document for different users or user groups.

### **active**

A window, dialog box, tabbed sheet or sheet object that is currently in use is called active or current.

### **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 field, but not by selections made in other fields. 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.

## **ASCII**

American Standard Code for Information Interchange. A standardized set of characters for computers.

## **B**

---

### **biff**

The native Excel file format.

### **blue**

Color of a cell that is locked.

### **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.

## **C**

---

### **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.

**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 "&".

**Container**

The container is an object that contains other objects. The container can contain all other sheet objects. The objects are grouped together and have common settings for font, layout and caption.

**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.

**CSV**

CSV is a file extension for comma separated value files. Used when a table is stored as a text file.

**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.

**D**

---

**Data source**

Data source is a name for a physical database, defined through the ODBC interface.

## **DBMS**

DBMS (DataBase Management System). The database program used in the storage and modification of data in the database.

## **delimiter**

A delimiter is 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 that 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 driver is 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.

## **E**

---

### **edit script dialog**

The text editor where the load script is edited.

### **excluded**

An excluded field value 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.

**F**

---

**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 an other.

**formatting**

Dual fields need formatting and often this needs to be explicitly done using the format() function.

**G**

---

**generic**

A QlikView script prefix used to unpack and load generic databases.

**gray**

Color of a cell that is excluded.

**green**

Color of a cell that has been selected.

**H**

---

**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.

## I

---

### **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.

## J

---

### **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.

## K

---

### **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.

## L

---

### **line/arrow object**

A sheet object used for adding lines or arrows to the layout. Line/arrow objects can be moved around and positioned anywhere in the sheet area, even to areas covered by other sheet objects.

**list box**

A sheet object that displays a list of field values. When a list is too long to display all choices, it will have a scroll bar, so that you can view additional items (see also field and value).

**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.

**locked**

A field value can be locked so that it does not become deselected by mistake. The layout can also be locked so that it is not altered by mistake.

**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. They are stored in the QlikView document.

---

**M**

---

**mapping**

A QlikView script prefix that enables the script developer to use lookup tables.

**module**

A dialog where your VB Script macros can be entered, tested and stored.

**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.

---

**N**

---

**nested aggregation**

A calculation using a two-step aggregation. See aggr function.

---

**O**

---

**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.

## **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.

## **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.

## **ooxml**

File format for Excel 2007 files (OpenXML).

## **optional**

An optional field value 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

## **P**

---

### **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**

The memory that the computer uses for programs that are running. Usually in the form of RAM.

## **Q**

---

### **QlikView (.qvw) file**

A QlikView (.qvw) file saved in binary format has the default extension qvw. Also called document.

### **QlikView AccessPoint**

Single point of access to corporate QlikView documents. Requires QlikView Server.

### **QlikView AJAX zero footprint client**

A lightweight object-based client to the QlikView Server that is based on the AJAX technology.



**QlikView Desktop**

QlikView with a bought user license, can be used to create documents and also to open documents created by other users.

**QlikView OCX**

QlikView packaged as an ActiveX component for integration into third party software. Not to be confounded with OCX replacement controls.

**QlikView Personal Edition**

A QlikView Desktop that 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.

**QlikView Plug-In**

Short for QlikView Desktop for Microsoft Internet Explorer.

**QlikView Publisher**

Manages content, access, and distribution of QlikView documents. The QlikView service and user interface are fully integrated in the QlikView Server (but an additional license is required).

**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.

**R**

---

**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.

**red**

Color of a cell that has been explicitly excluded by the user.

**relative path**

The location of a file, folder or directory on a disk, as seen in relation to 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 document. This can be done manually or automatically using a scheduled task or the QlikView Publisher.

**report**

With "report" one usually refers to a static paper or pdf 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: USERID, PASSWORD, SERIAL ACCESS, NTNAME, NTDOMAINSID, NTSID and OMIT.

**resident**

A table that has been loaded in the script can be accessed using a Load ... resident statement in the script.

**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.

**S**

---

**scalar function**

A scalar function is a function that operates on a single value to return a single scalar value. Examples: chr( ), sin( ), applymap( ). 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.

**Search Object**

Search objects can be used for searching for information anywhere in the document.

**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**

A QlikView script prefix used to load semantic links. These are used for linking one selection to another, e.g. self-references within a field.

**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/calendar object**

A sheet object which can be used to select values in a field or set values to one or two variables.

**SQL**

SQL, Structured Query Language, is 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 as well as several expressions with aggregated data, and has 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**

System variables are parameters generated by QlikView in the script for special purposes, e.g. definitions of document number formats

**T**

---

**tab in script**

The script can be split up onto several tabs. This improves the structure and facilitates navigation in the script.

**tabbed sheet**

See sheet.

**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.

**table file**

In QlikView, Table file means a text file representing a table in which the fields are separated by e.g. commas, tabs or semicolons. The content of the first row usually represents the names of the fields.

**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.

**U**

---

**URL**

Universal Resource Locator. A general address of the type used on the World Wide Web.

**V**

---

**value**

Fields are the primary data-carrying entity in QlikView. A field typically contains a number of values, called field values. Field values consist of numeric and/or alpha-numeric (text) data. See dual fields.

**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.

**W**

---

**wizard**

A series of dialogs to help the user achieve a given goal, e.g. creating a chart.

**X**

---

**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.



---

# Index

-	-	296, 788
	\$	
\$		282
\$fields		380
\$info		380
\$rows		380
\$table		380
\$variable		379
	&	
& (string operator)		296, 788
	*	
*		296, 788
	/	
/		296, 788
	+	
+		296, 788
	<	
<		297, 788
<=		297, 788
<>		297, 788
	=	
=		297, 788
	>	
>		297, 788
>=		297, 788

## A

above	837, 862
Access Restriction	195, 417
Access Restriction Table Wizard	195
Access Restrictions	417
Accumulation	637
acos	325
Actions	552-553
Add	222
Add Action dialog	553
Add Bookmark	117
Add Bookmark(s) (Bookmark Object)	60
AddMonths	353
Advanced aggregations	301
Advanced Color Map	653
Advanced Field Settings	515
Advanced search	108
after	840
age	365
aggr	301
Aggregation functions	298
Alert wizard	146
Alerts	141-142
Alias	223
Alignment	564
alt	340
Alternate States	113
Always show design menu items	78
AND	296, 788
And-mode in List Boxes	110
Animation dialog	623
Append only	411
Apply themes to other objects	603
applymap	337
ARGB	375
as	254
asin	325
atan	325
atan2	326
attribute	346
autogenerate	254



---

Automation	893
Automation and Macros	890
AutoNumber	332
AutoNumberHash128	332
AutoNumberHash256	332
Available Documents	57
avg	302, 794
Axes	663, 671, 679, 700, 732

**B**

Back	61
Backus-Naur Formalism	927
Bar Chart	617
Bar Chart: Object Menu	617
before	839
below	838
Binary	223
Bit operators	298, 789
bitcount	320, 830
Black	375
Black and Schole	330
BlackAndSchole	329
Block chart	713
Block Chart: Object Menu	713
Blue	376
Bookmark Object	579
Bookmark Object Properties: General	580
Bookmark Object: Object Menu	579
Bookmark Toolbar	52
Bookmarks	117, 119
Bookmarks menu	55
bottom	839
Box plot	765-766
Boxplot	765-766
Brown	376
Buffer	223
Bundle	224
Button	549
Button Properties: Actions	552-553
Button Properties: General	550
Button: Object Menu	549

## C

calculated dimensions	622
Call	225
Capitalize	335
Caption	459, 496, 506, 519, 532, 537, 547, 559, 566, 569, 577, 582, 588, 593, 597, 656, 664, 671, 679, 691, 700, 706, 712, 718, 726, 733, 746, 756
Caption and Border	458, 498
CD	283
ceil	316, 827
Cell Condition	211
Character set	257, 407
Chart Aggregation functions	790
Chart Color functions	375
Chart expression	757
chart expressions	785
Chart Field Functions	845
Chart Inter record functions	836
Chart Print...	59
Chart Properties	
Expressions	629
Chart Properties: Axes	649
Chart Properties: Axes(Scatter Chart)	689
Chart Properties: Colors	652
Chart Properties: Dimension Limits	625
Chart Properties: Dimensions	620
Chart Properties: Expressions (Scatter Chart)	684
Chart Properties: General	613
Chart Properties: Number	655
Chart Properties: Presentation (Bar Chart)	643, 661, 668, 676, 730
Chart Properties: Presentation (Block Chart)	716
Chart Properties: Presentation (Combo Chart)	643, 661, 668, 676, 730
Chart Properties: Presentation (Funnel Chart)	710
Chart Properties: Presentation (Gauge Chart)	722
Chart Properties: Presentation (Grid Chart)	685, 696
Chart Properties: Presentation (Line Chart)	643, 661, 668, 676, 730
Chart Properties: Presentation (Mekko Chart)	643, 661, 668, 676, 730
Chart Properties: Presentation (Pie Chart)	704
Chart Properties: Presentation (Pivot Table)	742
Chart Properties: Presentation (Radar Chart)	643, 661, 668, 676, 730
Chart Properties: Presentation (Scatter Chart)	685, 696
Chart Properties: Presentation (Straight Table)	752

---

Chart Properties: Sort	639
Chart Properties: Sort (Straight Table)	751
Chart Properties: Style	530, 745, 755
Chart Properties: Style (Bitmap Charts)	640
Chart Properties: Visual Cues	744, 754
Chart Range functions	830
Chart Ranking functions	843
Chart texts	648
Chart types	615
Chi2-test	781
chi2test_chi2	307, 806
chi2test_df	307, 806
chi2test_p	307, 805
chidist	326
chiinv	326
chr	333
Circular References	111
class	341
Clear	61
Clear All	61
Clipboard Copy List	96
Collaboration Pane	599
Color	375
Color Area	428
Color code	99
Color scheme	99
Colormix Wizard	760
ColorMix1	377
ColorMix2	377
Colors	663, 671, 679, 691, 700, 705, 711, 718, 725, 732
Column	843
ColumnNo	842
combin	319, 829
Combo Chart	665
Combo Chart: Object Menu	665
Comment field	226
Comment table	226
Comments	472
computername	342
concat	300, 790, 793
Concatenate	227
Conditional functions	340

---

Configure Toolbars	64
Connect	228
Connect to Data Source	188
Connect to Server	38
Connection pseudo URL	38
ConnectionString	346
Container	589
Container Object Properties: General	590
Container object: Object Menu	589
Container Properties	
Presentation	592
Context cell expansion	212
Context interpretation	213
ConvertToLocalTime	365
Copy mode	96
correl	302, 801
cos	325
cosh	326
count	300, 790, 793
Counter functions	331
Create Bookmark	117
Create Chart	43
Create Quick Chart	60
Create select statement	189
Crosstable	214, 229
Current Selections	103
Current Selections Box	533
Current Selections Box Properties: General	535
Current Selections Box: Object menu	533
Custom Error Messages	875
Custom Format Cell Dialog	524
Custom Object	595
Custom Object Properties: General	596
Custom Object: Object Menu	595
Customize Toolbars	64
Customizing QlikView toolbars in large deployments	66
Cyan	376
<b>D</b>	
DarkGray	375

---

---

data from files	160
Data Source	198
Data sources	160
Data Viewer	183
database	160
date	371
Date and time functions	350
date#	849
dateformat	287, 368
day	351
DayEnd	364
DaylightSaving	354
DayName	364
DayNames	288, 369
DayStart	364
Debugger	217
decimalsep	285, 367
Define expression	757
Definition	632
Delimiter	258
Design Toolbar	52
Dimension	620
Dimension Limits	625, 660, 667, 675, 683, 695, 703, 709, 715, 729, 750
dimensionality	837, 864
Dimensions	230, 660, 667, 675, 683, 695, 703, 709, 715, 721, 729, 741, 750
Direct Discovery	161
DirectCacheSeconds	234
DirectConnectionMax	234
DirectDateFormat	287, 368
DirectDistinctSupport	235
DirectFieldColumnDelimiter	235
DirectIdentifierQuoteStyle	235
DirectIdentifierQuoteChar	235
DirectMoneyDecimalSep	286, 367
DirectMoneyFormat	286, 368
Directory	236
DirectStringQuoteChar	235
DirectTableBoxListThreshold	235, 499
DirectTimeFormat	287, 368
DirectTimeStampFormat	287, 368
DirectUnicodeStrings	234
disconnect	236

distinct	253, 270
distinctrow	270
div	316, 826
Do..loop	236
Document data source	198
Document Functions	350
Document Properties	424
Scrambling	451
Server	434
Document Properties: Caption	459, 496
Document Properties: Extensions	452
Document Properties: Font	80, 453, 491
Document Properties: General	424
Document Properties: Groups	442
Document Properties: Layout	454, 492
Document Properties: Number	449
Document Properties: Opening	430
Document Properties: Presentation	448
Document Properties: Security	438
Document Properties: Sheets	432
Document Properties: Sort	447
Document Properties: Tables	444
Document Properties: Triggers	440
Document Properties: Variables	436-437, 544
Document Reports	127
Document Support Information	58
Dollar-sign expansion	290
Drag and drop to Microsoft Office	472
Drill to Details	499
Drop Field	237
Drop Table	237
dual	373, 851

**E**

e	330
Edit expression	757
Edit menu	51
Edit Module	895
Edit script	60
Edit Script dialog box	175
Embedded image files	157

---

embedded labels	259
Environmental Functions	342
Error variables	288
ErrorMode	289
evaluate	336
even	319, 830
Examples of Aggregate Qualifiers	853
Examples of Chart Inter Record functions	859
Examples of Chart Ranking functions	857
Examples of Concat Functions	854
Execute	238
exists	338
Exit script	238
exp	325
explicit labels	259
Exponential and logarithmic functions	325
ExponentNumberNotation	288, 369
Export	97
Export Action Settings	558
Export Bookmark	121
Export Contents	97
Expression Overview	155
Expression syntax for calculated formulas	871
Expressions	629, 660, 668, 676, 695, 703, 710, 716, 721, 729, 741, 750

## F

fabs	317, 828
fact	319, 829
false	330
fdist	327
Field	171
Field Functions (Chart)	845
Field tags	171
FieldIndex	340, 837
FieldName	349
FieldName()	349
FieldNumber	349
FieldNumber()	349
FieldValue	339, 836
fieldvaluecount	332, 843
File functions	346

---

---

File menu	47
File Print	59
File Wizard	197-203, 209, 216
File Wizard: Advanced	209
File Wizard: Source	197
File Wizard: Transform	203
File Wizard: Type	198
filebasename	346
filedir	346
fileextension	347
filename	346
Filename	347
FileName	347
FilePath	347
FileSize	347
filetime	347
Fill Cells	211
Financial aggregation functions in charts	803
Financial aggregation functions in script	306
Financial functions	328
Find/replace	182
findoneof	336
finv	327
First	238, 841
firstsortedvalue	299, 790, 792
FirstValue	300
firstworkdate	365
floor	317, 827
floppy	283
fmod	316, 827
Font	80, 453, 491, 506, 518, 532, 537, 547, 559, 566, 577, 582, 587, 593, 656, 664, 671, 679, 691, 700, 705, 712, 718, 725, 733, 746, 756
Font (Chart Title)	80, 453, 491
For each..next	240
For..next	239
Force	241
Format	490
Format codes	885
Format specification items	257
Formatting functions	370
Forward	61
frac	317, 827

---



fractile	302, 802
Free text searches	105
from	254
Funnel chart	707
Funnel chart: Object Menu	707
FV	328

## G

Gauge Chart	719
Gauge Chart: Object Menu	719
General	619, 660, 667, 675, 683, 709, 715, 721, 729, 741, 750
General numeric functions	316
General numeric functions (Chart)	826
Generic	242
getalternativecount	343, 847
getcurrentfield	342, 845
getcurrentselections	343, 846
getexcludedcount	343, 847
getfieldselections	343, 845
getfolderpath	348
getnotselectedcount	343, 846
getpossiblecount	343, 846
GetRegistryString	344
getselectedcount	343, 846
Getting Started Wizard	43
GMT	354
Green	376
Grid Chart	693
Grid Chart: Object Menu	693
group by	255, 271
Group Settings	443

## H

Has Error Bars	629
hash128	336
Hash160	336
Hash256	337
having	271
Header size	259
Help menu	57
Hidden script	60, 183

---

hideprefix	283
hidesuffix	283
Hierarchic groups (drill-down)	873
Hierarchy	242
Hierarchy Parameters	215
HierarchyBelongsTo	243
hour	351
HRank	844
HSL	375

**I**

if	340
If..then..elseif..else..end if	244
Image stretch	564
Image_size	245
Import Bookmark	122
include	283
InDay	358
InDayToTime	358
Independent Samples T-test	783
index (string function)	333-334
Indicator	101
Info	245
inline	254
Inline Data Wizard	194
InLunarWeek	357
InLunarWeekToDate	358
InMonth	356
InMonths	356
InMonthsToDate	356
InMonthToDate	356
Inner	245
Input	345
Input Box	539
Input Box Properties	
Presentation	543
Input Box Properties: Constraints	436, 544
Input Box Properties: General	541
Input Box Properties: Number	546
Input Box: Object Menu	539
Input field	246

Input field aggregation function	817
Input Fields	172
inputavg	818
Inputfield	246
inputsum	818
InQuarter	355
InQuarterToDate	355
Insert and Update	413
Insert only	412
Insert Update and Delete	414
Inter record functions	338
Internal Files	157
Internal Macro Interpreter	893
Internal VBScript Interpreter	893
Interpretation functions	847
interval	373
interval#	850
IntervalMatch	247, 249
InWeek	357
InWeekToDate	357
InYear	355
InYearToDate	355
IRR	306, 803
isnull	342
isnum	341
IsPartialReload	342
istext	342
Item Settings	139
iterno	332

**J**

Join	250, 270-271
------	--------------

**K**

Keep	250
keepchar	335
Keyboard Command Shortcuts	907
kurtosis	298, 302, 802

**L**

Label	633
-------	-----

---

Labels	259
last	841
LastValue	300
lastworkdate	365
Launch	552
Layout	454, 492, 506, 518, 532, 537, 547, 559, 566, 569, 577, 582, 587, 593, 597, 656, 664, 671, 679, 691, 700, 706, 712, 718, 726, 733, 746, 756
Layout menu	53
Left	251, 333
Legend Settings	646
len	333
Let	252
LightBlue	376
LightCyan	376
LightGray	376
LightGreen	376
LightMagenta	376
LightRed	376
Limited Load	218
Line Chart	657
Line Chart: Object Menu	658
Line/Arrow Object	567
Line/Arrow Object Properties: General	568
Line/Arrow Object: Object Menu	567
linest	303
linest_b	303, 796
linest_df	305, 799
linest_f	305, 799
linest_m	303, 796
linest_r2	304, 797
linest_seb	304, 798
linest_sem	304, 797
linest_sey	304, 798
linest_ssreg	305, 800
linest_ssresid	306, 800
Linked tables	161
List Box	473
Tree View	478, 498
List box as straight table	479
List Box Properties	
Expressions	479
List Box properties: General	476

---

---

List Box Properties: Number	490
List Box Properties: Presentation	487
List Box Properties: Sort	486
List Box: Object Menu	473
Load	253
Load inline	194
loading data	160
Loading data	160
Local objects	471
LocalTime	352
Lock Selections	61
log	325
log10	325
Logical functions	341
Logical operators	296, 788
LongDayNames	288, 369
LongMonthNames	287, 369
Lookup	340
Loosely Coupled Tables	111, 925
Loosen Table	260
lower	334
ltrim	334
LunarweekEnd	363
LunarWeekName	363
LunarWeekStart	363

## M

Macro	290, 468, 893
Macros	440, 468, 552, 893
Macros on QlikView Server	903
Magenta	376
makedate	352
maketime	353
makeweekdate	353
Map ... using	261
Mapping	261
Mapping functions	337
MapSubstring	338
match	341
Mathematical constants and parameter-free functions	330

max	299, 790-791
MaxString	300, 790, 793
median	303, 801
Mekko chart	727
Mekko Chart: Object Menu	727
mid	333
min	299, 790-791
Mini chart settings	484
Minstring	790
MinString	300, 792
minute	350
missingCount	790
MissingCount	301, 794
mixmatch	341
mod	316, 826
mode	299, 790, 792
money	371
money#	848
moneydecimalsep	286, 367
MoneyFormat	286, 367
moneythousandsep	286, 367
month	351
MonthEnd	360
MonthName	361
MonthNames	287, 369
MonthsEnd	361
MonthsName	361
MonthsStart	361
MonthStart	360
Moving selections	102
MsgBox	344
msq	260
Multi Box	507
Multi Box Properties: General	510
Multi Box Properties: Number	517
Multi Box Properties: Presentation	513
Multi Box Properties: Sort	512
Multi Box: Object Menu	507
Multiple selection within field	101
must_include	283

## N

Navigation Toolbar	52
Nested aggregations	865
networkdays	365
New Chart	43, 610, 617, 657
New file	59
New Variable	437, 544
no eof	258
no labels	259
no quotes	260
NoConcatenate	262
Non-hierarchic groups (cyclic)	874
NoOfColumns	842
NoOfFields	349
NoOfFields()	349
NoOfReports	350
NoOfRows	349, 842, 863
NoOfRows()	349
NoOfTables	349
NoOfTables()	349
normdist	326
norminv	327
not	296, 788
Notes	472
now	352
nPer	328
NPV	306, 804
null	342
Null functions	342
NullAsNull	263
NullAsValue	262
nullcount	790
NullCount	301, 794
nulldisplay	284
nullinterpret	285
nullvalue	285
num	370
num#	847
numavg	318, 828
Number	490, 664, 671, 679, 691, 700, 705, 711, 718, 725, 733, 746, 756
Number Format	449

Number interpretation variables	285, 366
Numbers on Data Points	629
numcount	318, 828
Numeric operators	296, 788
numericcount	790
NumericCount	301, 793
nummax	318, 829
nummin	318, 829
numsum	317, 828

**O**

Object Caption Style	426
Object menu	56, 471
odd	319, 830
only	299, 790, 792
Open Database Files	192
Open file	59
Open Internet Files	193
Operators	295, 787
OR	296, 788
ord	333
order by	255, 271
osuser	342
othersymbol	285
Outer	263

**P**

Page Settings	136
Paired T-test	782
peek	339
permut	319, 830
Personal Edition	25
pi	330
pick	340
Pie Chart	701
Pie Chart: Object Menu	701
Pivot Table	735
Pivot Table: Object Menu	738
Pmt	328
Pop-up Window Settings	427
pow	325



---

Predefined values	438, 545
previous	339
Print	59
Print as PDF	60
Print as PDF (Object)	60
Print Options: Date & Time	95
Print Options: General	91
Print Preview	96
Print: General	91
Print: Header/Footer	93
Print: Layout	92
Project Files	45
Properties files	45
purgechar	335
PV	329

## Q

QlikTechBlue	377
QlikTechGray	377
QlikView Desktop	25
QlikView file	45
QlikView Personal Edition	25
QlikViewProject Files	45
qlikviewversion	344
Qualify	264
QuarterEnd	360
QuarterName	360
QuarterStart	359
Quick Chart Wizard	767
Quotation marks	292
Quotes	260
QVD files	409
QVD files and Incremental Load	410
QvdCreateTime	348
QvdFieldName	348
QvdNoOfFields	348
QvdNoOfRecords	348
QvdTableName	348
QVPath	283
QVRoot	284
qvuser	342

---

QVWorkPath	284
QVWorkRoot	284

**R**

Radar Chart	673
Radar Chart: Object Menu	673
rand	330
Range functions (Chart)	830
Range functions (Script)	320
RangeAvg	320, 831
RangeCorrel	323, 836
RangeCount	320, 831
RangeFractile	321, 833
RangeIRR	323
RangeKurtosis	321, 833
RangeMax	321, 832
RangeMaxString	322, 835
RangeMin	321, 831
RangeMinString	322, 835
RangeMissingCount	322, 834
RangeMode	323, 835
RangeNPV	324
RangeNullCount	322, 834
RangeNumericCount	322, 833
RangeOnly	323, 836
RangeSkew	321, 832
RangeStdev	321, 832
RangeSum	320, 831
RangeTextCount	322, 833
RangeXIRR	324
RangeXNPV	324
rank	843
Rate	329
recno	331
Record length	259
Red	376
Redo Layout	60
Reexecute script	60
Reference Lines	647, 687, 699
Refresh	59
Relational operators	297, 788

---

Reload	60
reloadtime	342
Rem	265
Rename Field	265
Rename Table	266
Renaming fields	384
repeat	334
Replace	268, 336
Report	126
Report Editor	126
Report Settings	133
ReportComment	350
ReportID	350
ReportName	350
ReportNumber	350
Reports	125
Reports menu	55, 125
Reset Clear State	61
resident	254
RGB	375
Right	267, 333
round	317, 827
rowno	331, 842, 863
rtrim	334

**S**

Sample	269
Scatter Chart	681
Scatter Chart: Object Menu	681
Scrambling	451
Script Color functions	375
Script control statements	221
Script Debugger	217
Script expressions	295
Script management	175
Script Password	417
Script statements and keywords	219
Script syntax	219
Script variables	282
ScriptError	289
ScriptErrorCount	290

ScriptErrorDetails	290
ScriptErrorList	290
Scripting Engine	893
Search	60, 105, 108
Search Object	583
Object Menu	583
Search Object Properties	
General	584
Presentation	586
Sort	587
second	350
secondarydimensionality	837, 864
Section	269
Security	195, 417
security field	195
Select (SQL)	270
Select Fields	465
Selection state	99
Selections in charts and tables	611
Selections menu	53
Selections page	136
Semantic	272
Server objects	471
Server Objects Pane	599
Set	272, 282
Set Analysis	819
Set Clear State	61
Set Statement Wizard	181
SetDateYear	354
SetDateYearMonth	354
Settings menu	54
Sheet Properties	462
Sheet Properties: Fields	465
Sheet Properties: General	463
Sheet Properties: Objects	466
Sheet Properties: Security	467
Sheet Properties: Triggers	468
Sheet toolbar	52
Sheet: Object Menu	462
ShowCalendarWeek	288, 369
sign	320, 830
sin	325

---

sinh	326
Sizing and moving chart components	612
skew	298, 303, 802
Sleep	272
Slider/calendar object	571
Slider/Calendar Object Properties:	
Presentation (calendar mode)	576
Presentation (Slider Mode)	575
Slider/Calendar Object Properties: General	573
Slider/calendar object: Object Menu	571
Sort	660, 668, 676, 684, 696, 704, 710, 722, 729, 741
Sparkline	484, 629
Specify cells for new column	212
Specify Row Condition	210
SQL	272
SQLColumns	273
SQLTables	273
SQLTypes	274
sqr	325
sqrt	325
Standard Toolbar	52
Star	275
Start page	36
Statistical distribution functions	326
Statistical test functions in charts	805
Statistical test functions in script	307
Statistics Box	501
Statistics Box Properties: Number	505
Statistics Box Properties:General	503
Statistics Box: Object Menu	501
Statistics Chart Wizard	781
Status bar	67
stdev	298, 302, 801
stdv	302, 794
sterr	303, 795
steyx	303, 795
Store	276
Straight Table	747
Straight Table: Object Menu	747
String functions	333
String operators	296, 788
StripComments	284

Style	530, 660, 668, 676, 685, 696, 704, 710, 716, 722, 729, 745, 755
Sub..end sub	276
subfield	335
substringcount	337
sum	298, 790-791
Switch..case..default..end switch	277
Synthetic dimension functions	825
SysColor	375, 377
System fields	379
System Functions	342
System variables	283

## T

Table Box	521
Table Box Properties: General	525
Table Box Properties: Number	531
Table Box Properties: Presentation	528
Table Box Properties: Sort	527
Table Box: Object Menu	521
Table file	160
Table File Wizard: Type	198
Table format	257
Table functions	349
Table labels	282
Table Viewer	183
TableName	349
TableName()	349
TableNumber	350
TableNumber()	350
Tag	278
tan	325
tanh	326
tdist	327
text	851
Text Object	561
Text Object Properties: General	562
Text Object: Object Menu	561
Text on Data Point	629
TextBetween	336
textcount	790
TextCount	301, 794

---

Theme Maker Wizard	603-604
Themes	601
thousandsep	285, 367
time	372
Time Chart Wizard	775
time#	849
timeformat	286, 368
timestamp	374
timestamp#	850
timestampformat	287, 368
TimeZone	354
tinverse	327
Title Settings	616
today	352
Toolbars	52
Tools menu	55
top	838, 859
Trace	279
Transform Table	203
Tree View in list boxes	476, 498
Trellis	625
Trigonometric and hyperbolic functions	325
trim	335
true	330
TTest_conf	308, 808
TTest_df	307, 806
TTest_dif	308, 807
TTest_lower	308, 808
TTest_sig	308, 807
TTest_sterr	308, 807
TTest_t	307, 806
TTest_upper	308, 808
TTest1_conf	311, 812
TTest1_df	310, 811
TTest1_dif	311, 811
TTest1_lower	311, 812
TTest1_sig	310, 811
TTest1_sterr	311, 812
TTest1_t	310, 811
TTest1_upper	311, 812
TTest1w_conf	312, 814
TTest1w_df	312, 813

TTestlw_dif	312, 813
TTestlw_lower	312, 814
TTestlw_sig	312, 813
TTestlw_sterr	312, 814
TTestlw_t	311, 813
TTestlw_upper	313, 815
TTestw_conf	310, 810
TTestw_df	309, 809
TTestw_dif	309, 809
TTestw_lower	310, 810
TTestw_sig	309, 809
TTestw_sterr	309, 810
TTestw_t	309, 808
TTestw_upper	310, 810

## U

Undo Layout	60
Unless	279
Unlock	103
Unlock and Clear All	61
Unlock Selections	61
Unmap	280
Unqualify	280
Untag field	280
upper	334
User Preferences Editor	74
User Preferences: Design	77
User Preferences: Export	81
User Preferences: Folders	85
User Preferences: General	69
User Preferences: Mail	84
User Preferences: Objects	78
User Preferences: Print	83
User Preferences: Registration	88
User Preferences: Save	73
User Preferences: Security	87
User Reports	127
Using the Pivot Table	735
UTC	354



---

<b>V</b>	
Value handling variables	284
Variable	173
Variable Overview	153
Verbatim	284
View menu	52
VRank	845

<b>W</b>	
Waterfall chart	631
week	351
weekday	352
WeekEnd	362
WeekName	362
WeekStart	362
weekyear	351
when	281
where	254, 270
Where Clause	214
while	255
White	376
WildMatch	341
Window menu	56
WinPath	284
WinRoot	284

<b>X</b>	
XIRR	306, 803
XML	45
xml files	260
XNPV	306, 804
xor	296, 788

<b>Y</b>	
year	351
year2date	353
YearEnd	359
YearName	359
YearStart	358

Yellow 377

**Z**

ZTest\_conf 313, 816  
ZTest\_dif 313, 815  
ZTest\_sig 313, 815  
ZTest\_sterr 313, 816  
ZTest\_z 313, 815  
ZTestw\_conf 314, 817  
ZTestw\_dif 314, 817  
ZTestw\_sig 314, 816  
ZTestw\_sterr 314, 817  
ZTestw\_z 314, 816