# QlikView 

## Reference Manual

## Copyright © 1994-2011 Qlik®Tech International AB, Sweden.

Under international copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written permission of QlikTech International AB, except in the manner described in the software agreement.
Qlik ${ }^{\circledR}$ Tech and $\mathrm{Qlik}{ }^{\circledR}$ View are registered trademarks of QlikTech International AB.
Microsoft, MS-DOS, Windows, Windows NT, Windows 2000, Windows 2003, Windows XP, Windows Vista, Windows 2008, SQL Server, FoxPro, Excel, Access and MS Query are trademarks of Microsoft Corporation.
IBM, AS/400 and PowerPC are trademarks of International Business Machines Corporation.
Borland, Paradox and dBASE are trademarks of Borland International.
ORACLE and SQL*Net are trademarks of Oracle Corporation.
MacOS is a trademark of Apple Corporation.
Sun Java and JavaBeans are trademarks of Sun Microsystems, Inc.

## Content

Part 1 Introduction ..... 17
1 Before You Begin ..... 19
1.1 Conventions ..... 19
1.2 QlikTech Support Services ..... 20
2 What's New in QlikView 11? ..... 21
2.1 QlikView Desktop ..... 21
2.2 AJAX Client Improvements. ..... 22
2.3 Server Components. ..... 23
2.4 WebParts and WorkBench ..... 25
2.5 APIs ..... 25
3 Compatibility Issues for Earlier Versions of QlikView ..... 27
3.1 File Format Compatibility ..... 27
3.2 Mixed Client/Server Environments ..... 27
4 QlikView Personal Edition. ..... 29
Part 2 Installation ..... 31
5 Installing QlikView ..... 32
5.1 Starting the Installation Program ..... 32
5.2 When the Installation Program Is Running ..... 32
5.3 Serialization. ..... 33
5.4 Network Distribution of QlikView ..... 33
5.5 License Enabler File Editor. ..... 33
5.6 Updating QlikView ..... 34
6 OLE DB and ODBC ..... 35
6.1 OLE DB ..... 35
6.2 When Is the ODBC Interface Needed? ..... 35
6.3 Creating ODBC Data Sources ..... 36
Part 3 Navigation and Basics ..... 37
7 Starting QlikView ..... 38
7.1 Command Line Syntax. ..... 38
7.2 The Start Page. ..... 39
7.3 Connect to Server ..... 4
7.4 Batch Execution ..... 42
8 Getting Started Wizard ..... 45
9 QlikView File ..... 51
9.1 QlikView Project Files. ..... 51
9.2 Source Control ..... 52
10 Menu Commands ..... 53
10.1 File Menu. ..... 53
10.2 Edit Menu. ..... 57
10.3 View Menu. ..... 58
10.4 Selections Menu. ..... 59
10.5 Layout Menu ..... 59
10.6 Settings Menu. ..... 61
10.7 Bookmarks Menu. ..... 61
10.8 Reports Menu. ..... 62
10.9 Tools Menu. ..... 62
10.10 Object Menu. ..... 63
10.11 Window Menu. ..... 63
10.12 Help Menu. ..... 64
11 Toolbars and Status Bar ..... 67
11.1 Toolbars ..... 67
11.2 Standard Toolbar ..... 67
11.3 Navigation Toolbar ..... 69
11.4 Design Toolbar ..... 69
11.5 Sheet Toolbar ..... 72
11.6 Bookmark Toolbar ..... 72
11.7 Customize (Toolbars) ..... 72
11.8 Status Bar ..... 75
12 User Preferences ..... 77
12.1 User Preferences: General ..... 77
12.2 User Preferences: Save. ..... 81
12.3 User Preferences: Editor. ..... 82
12.4 User Preferences: Design. ..... 85
12.5 User Preferences: Objects ..... 86
12.6 Font ..... 88
12.7 User Preferences: Export ..... 89
12.8 User Preferences: Printing ..... 92
12.9 User Preferences: Mail ..... 93
12.10 User Preferences: Locations ..... 94
12.11 User Preferences: Security ..... 96
12.12 User Preferences: License. ..... 97
13 Exporting and Printing ..... 99
13.1 Print: General ..... 99
13.2 Print: Layout ..... 100
13.3 Print: Header/Footer ..... 101
13.4 Date \& Time. ..... 103
13.5 Print Sheet ..... 103
13.6 Print Preview ..... 104
13.7 Copy Mode. ..... 104
13.8 Export/Export Contents ..... 105
14 Logic and Selections ..... 107
14.1 Single Field Value Selection ..... 107
14.2 Color Scheme ..... 107
14.3 Selection Styles ..... 108
14.4 Indicator. ..... 109
14.5 Multiple Selection Within a Field. ..... 109
14.6 Moving Selections ..... 109
14.7 Locking Selected Field Values ..... 110
14.8 Current Selections. ..... 111
14.9 Selections in Other Objects. ..... 111
14.10 Search ..... 113
14.11 Advanced Search Dialog ..... 116
14.12 And-Mode in List Boxes ..... 117
14.13 Loosely Coupled Tables. ..... 118
14.14 Alternate States ..... 122
15 Bookmarks ..... 125
15.1 Bookmark Types ..... 125
15.2 Add Bookmark Dialog ..... 125
15.3 Bookmarks Dialog ..... 127
15.4 Export Bookmark(s) ..... 129
15.5 Import Bookmarks ..... 129
15.6 Alternate States and Bookmarks ..... 130
16 Reports ..... 133
16.1 Reports Menu. ..... 133
16.2 Report Editor ..... 134
17 Alerts ..... 149
17.1 Creating Alerts ..... 149
17.2 Using Alerts. ..... 149
17.3 Alerts ..... 150
17.4 The Alerts Dialog ..... 151
17.5 Alert Wizard ..... 154
18 Variable Overview ..... 161
19 Expression Overview ..... 163
20 Internal Files ..... 165
20.1 File References in the QlikView Layout ..... 165
20.2 Internal Files in QlikView Documents ..... 165
20.3 Internal Files in the QlikView Program File ..... 165
20.4 Sounds ..... 166
Part 4 Script ..... 167
21 Introduction to Loading Data ..... 168
21.1 Loading Data from Files ..... 168
21.2 Loading Data from Databases ..... 168
22 Variables and Fields ..... 169
22.1 Fields ..... 169
22.2 Input Fields ..... 170
22.3 Variable ..... 171
22.4 Other Entities in QlikView ..... 171
23 Edit Script Dialog ..... 173
23.1 Set Statement Wizard ..... 179
23.2 Find/Replace (Script) ..... 180
23.3 Hidden Script ..... 181
23.4 Table Viewer ..... 181
23.5 Data Link Properties ..... 184
23.6 Connect to Data Source. ..... 186
23.7 Create Select Statement ..... 187
23.8 Open Local Files ..... 190
23.9 Open Internet Files or Open QlikView Document. ..... 191
23.10 Inline Data Wizard. ..... 192
23.11 Access Restriction Table Wizard. ..... 193
23.12 File Wizard. ..... 194
24 Script Syntax ..... 219
24.1 Script Statements and Keywords. ..... 219
24.2 Table Names ..... 272
24.3 Script Variables. ..... 273
24.4 Dollar-Sign Expansions. ..... 279
24.5 Quotation Marks ..... 281
25 Script Expressions ..... 283
25.1 Operators. ..... 283
25.2 Aggregation Functions ..... 286
25.3 Other Functions. ..... 303
26 Data Structures. ..... 365
26.1 Data Loading Statements ..... 365
26.2 Execution of the Script ..... 365
26.3 System Fields. ..... 365
26.4 Logical Tables. ..... 366
26.5 Associations between Logical Tables. ..... 367
26.6 Renaming Fields ..... 369
26.7 Concatenating Several Tables into One ..... 369
26.8 Join and Keep ..... 370
27 Evaluating the Loaded Data ..... 375
27.1 Generic Databases. ..... 375
27.2 Cross Tables. ..... 376
27.3 Matching Intervals to Discrete Data ..... 377
27.4 Using the Extended IntervalMatch Syntax to Resolve Slowly Changing Dimension Problems ..... 379
27.5 Hierarchies ..... 380
27.6 Semantic Links ..... 382
27.7 Linking Information to Field Values ..... 385
27.8 Data Cleansing ..... 386
27.9 Wild Cards in the Data ..... 387
27.10 NULL Value Handling in QlikView. ..... 388
27.11 Character Set ..... 391
28 QVD Files. ..... 393
28.1 Purpose of QVD Files ..... 393
28.2 Create QVD Files. ..... 393
28.3 Reading Data from QVD Files ..... 394
28.4 Using QVD Files for Incremental Load ..... 394
29 Security ..... 401
29.1 Authentication and Authorization. ..... 401
29.2 Security Using the QlikView Publisher ..... 401
29.3 Security Using the Section Access in the QlikView Script ..... 401
29.4 Sections in the Script ..... 401
29.5 Access Levels in Section Access. ..... 402
29.6 Section Access System Fields. ..... 402
29.7 Restrictions on QlikView Functionality. ..... 404
29.8 Dynamic Data Reduction. ..... 404
29.9 Inherited Access Restrictions. ..... 405
29.10 Encryption. ..... 405
Part 5 Sheet and Sheet Objects ..... 407
30 Document Properties. ..... 408
30.1 Document Properties: General. ..... 408
30.2 Document Properties: Opening ..... 414
30.3 Document Properties: Sheets ..... 416
30.4 Document Properties: Server ..... 418
30.5 Document Properties: Scheduler ..... 420
30.6 Input Box Properties: Constraints and Document Properties: Variables. ..... 421
30.7 Document Properties: Security ..... 423
30.8 Document Properties: Triggers. ..... 425
30.9 Document Properties: Groups. ..... 427
30.10 Document Properties: Tables ..... 429
30.11 Document Properties: Sort. ..... 432
30.12 Document Properties: Presentation ..... 433
30.13 Document Properties: Number. ..... 434
30.14 Document Properties: Scrambling ..... 436
30.15 Font ..... 437
30.16 Layout ..... 438
30.17 Caption ..... 442
31 The Sheet ..... 445
31.1 Creation ..... 445
31.2 Navigation ..... 445
31.3 Sheet: Object Menu ..... 446
31.4 Sheet Properties. ..... 446
31.5 Sheet Properties: General ..... 447
31.6 Select Fields/Sheet Properties: Fields. ..... 449
31.7 Sheet Properties: Objects, ..... 450
31.8 Sheet Properties: Security. ..... 451
31.9 Sheet Properties: Triggers ..... 452
31.10 Tabrow Properties. ..... 453
32 Sheet Objects ..... 455
32.1 Local Objects and Server Objects. ..... 455
32.2 Object Menu ..... 456
32.3 Drag and Drop to Microsoft Office ..... 456
33 List Box ..... 457
33.1 List Box: Object Menu. ..... 457
33.2 List Box Properties: General ..... 460
33.3 List Box Properties: Expressions. ..... 463
33.4 List Box Properties: Sort ..... 470
33.5 List Box Properties: Presentation. ..... 471
33.6 Properties: Number. ..... 473
33.7 Font ..... 475
33.8 Layout ..... 476
33.9 Caption. ..... 479
33.10 List Box Tree View ..... 481
34 Statistics Box ..... 483
34.1 Statistics Box: Object Menu ..... 483
34.2 Statistics Box Properties: General ..... 485
34.3 Statistics Box Properties: Number ..... 487
34.4 Font ..... 488
34.5 Layout ..... 488
34.6 Caption ..... 488
35 Multi Box ..... 489
35.1 Multi Box: Object Menu ..... 489
35.2 Multi Box Properties: General ..... 491
35.3 List Box Properties: Sort, ..... 493
35.4 Multi Box Properties: Presentation ..... 494
35.5 Multi Box Properties: Number. ..... 498
35.6 Font ..... 499
35.7 Layout ..... 499
35.8 Caption ..... 500
36 Table Box ..... 501
36.1 Using the Table Box ..... 501
36.2 Table Box: Object Menu ..... 501
36.3 Table Box Properties: General. ..... 505
36.4 Table Box Properties: Sort ..... 507
36.5 Table Box Properties: Presentation. ..... 508
36.6 Chart Properties: Style. ..... 510
36.7 Table Box Properties: Number. ..... 511
36.8 Font ..... 512
36.9 Layout ..... 512
36.10 Caption ..... 512
37 Current Selections Box ..... 513
37.1 Current Selections Box: Object menu. ..... 513
37.2 Current Selections Box Properties: General ..... 515
37.3 Font ..... 516
37.4 Layout ..... 516
37.5 Caption ..... 517
38 Input Box ..... 519
38.1 Using the Input Box ..... 519
38.2 Input Box: Object Menu ..... 519
38.3 Input Box Properties: General ..... 521
38.4 Input Box Properties: Presentation. ..... 523
38.5 Input Box Properties: Constraints and Document Properties: Variables. ..... 524
38.6 Input Box Properties: Number ..... 526
38.7 Font. ..... 527
38.8 Layout ..... 527
38.9 Caption ..... 527
39 Button ..... 529
39.1 Button: Object Menu. ..... 529
39.2 Button Properties: General ..... 530
39.3 Actions. ..... 532
39.4 Font ..... 537
39.5 Layout ..... 538
39.6 Caption ..... 538
40 Text Object ..... 539
40.1 Text Object: Object Menu. ..... 539
40.2 Text Object Properties: General ..... 540
40.3 Actions ..... 543
40.4 Font ..... 544
40.5 Layout ..... 544
40.6 Caption. ..... 544
41 Line/Arrow Object ..... 545
41.1 Line/Arrow Object: Object Menu ..... 545
41.2 Line/Arrow Object Properties: General ..... 546
41.3 Actions. ..... 547
41.4 Layout ..... 547
41.5 Caption ..... 547
42 Slider/Calendar Object ..... 549
42.1 Using a Slider/Calendar Object. ..... 549
42.2 Slider/Calendar Object: Object Menu. ..... 549
42.3 Slider/Calendar Object Properties: General ..... 551
42.4 Slider/Calendar Object Properties: Presentation (Slider Mode). ..... 553
42.5 Slider/Calendar Object Properties: Presentation (Calendar Mode). ..... 554
42.6 Sort ..... 555
42.7 Number ..... 555
42.8 Font ..... 555
42.9 Layout ..... 555
42.10 Caption ..... 555
43 Bookmark Object ..... 557
43.1 Bookmark Objects in Alternate States ..... 557
43.2 Bookmark Object: Object Menu ..... 557
43.3 Bookmark Object Properties: General ..... 558
43.4 Font ..... 560
43.5 Layout ..... 560
43.6 Caption. ..... 560
44 Search Object. ..... 561
44.1 Search Object: Object Menu. ..... 561
44.2 Search Object Properties: General ..... 562
44.3 Search Object Properties: Presentation. ..... 564
44.4 Search Object Properties: Sort ..... 565
44.5 Font ..... 565
44.6 Layout ..... 565
44.7 Caption ..... 566
45 Container ..... 567
45.1 Container: Object Menu. ..... 567
45.2 Container Properties: General ..... 568
45.3 Container Properties: Presentation ..... 570
45.4 Font ..... 571
45.5 Layout ..... 571
45.6 Caption. ..... 571
46 Custom Object ..... 573
46.1 Custom Object: Object Menu. ..... 573
46.2 Custom Object Properties: General ..... 574
46.3 Layout ..... 575
46.4 Caption. ..... 575
47 Server Objects Pane ..... 577
47.1 Configuring the Pane. ..... 577
47.2 Shared Objects ..... 577
47.3 Adding and Sharing Objects ..... 577
47.4 Server Objects Dialog. ..... 578
48 Layout Themes ..... 581
48.1 About QlikView Layout Themes. ..... 581
48.2 Applying Themes in the Layout ..... 582
48.3 Theme Maker Wizard. ..... 583
Part 6 Charts ..... 587
49 Introduction ..... 588
49.1 New Chart ..... 590
49.2 Selections in Charts and Tables. ..... 591
49.3 Fast Chart Type Change ..... 592
49.4 Sizing and Moving Chart Components. ..... 592
49.5 Chart Properties. ..... 592
49.6 Chart Properties: General ..... 593
50 Bar Chart ..... 597
50.1 New Chart ..... 597
50.2 Bar Chart: Object Menu. ..... 597
50.3 General ..... 599
50.4 Chart Properties: Dimensions. ..... 600
50.5 Chart Properties: Dimension Limits ..... 605
50.6 Chart Properties: Expressions ..... 609
50.7 Chart Properties: Sort ..... 618
50.8 Chart Properties: Style. ..... 619
50.9 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart), ..... 622
50.10 Chart Properties: Axes (Bar- Line- Combo- Radar- Mekko Chart). ..... 628
50.11 Chart Properties: Colors. ..... 631
50.12 Chart Properties: Number. ..... 634
50.13 Font ..... 635
50.14 Layout ..... 635
50.15 Caption. ..... 635
51 Line Chart ..... 637
51.1 New Chart ..... 637
51.2 Line Chart: Object Menu ..... 638
51.3 General ..... 640
51.4 Dimensions ..... 640
51.5 Dimension Limits. ..... 640
51.6 Expressions. ..... 640
51.7 Sort ..... 640
51.8 Style. ..... 640
51.9 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart) ..... 641
51.10 Axes. ..... 643
51.11 Colors ..... 643
51.12 Number ..... 644
51.13 Font ..... 644
51.14 Layout ..... 644
51.15 Caption. ..... 644
52 Combo Chart ..... 645
52.1 Combo Chart: Object Menu. ..... 645
52.2 General ..... 647
52.3 Dimensions ..... 647
52.4 Dimension Limits. ..... 647
52.5 Expressions ..... 648
52.6 Sort ..... 648
52.7 Style ..... 648
52.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart) ..... 648
52.9 Axes. ..... 651
52.10 Colors ..... 651
52.11 Number ..... 651
52.12 Font ..... 651
52.13 Layout ..... 651
52.14 Caption. ..... 651
53 Radar Chart ..... 653
53.1 Radar Chart: Object Menu ..... 653
53.2 General ..... 655
53.3 Dimensions ..... 655
53.4 Dimension Limits. ..... 655
53.5 Expressions ..... 656
53.6 Sort ..... 656
53.7 Style. ..... 656
53.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart) ..... 656
53.9 Axes. ..... 659
53.10 Colors. ..... 659
53.11 Number ..... 659
53.12 Font ..... 659
53.13 Layout ..... 659
53.14 Caption. ..... 659
54 Scatter Chart ..... 661
54.1 Scatter Chart: Object Menu. ..... 661
54.2 General ..... 663
54.3 Dimensions ..... 663
54.4 Dimension Limits ..... 663
54.5 Chart Properties: Expressions (Scatter Chart). ..... 664
54.6 Sort ..... 664
54.7 Style. ..... 665
54.8 Chart Properties: Presentation (Scatter- Grid Chart) ..... 665
54.9 Chart Properties: Axes (Scatter- Grid Chart). ..... 669
54.10 Colors ..... 670
54.11 Number ..... 671
54.12 Font ..... 671
54.13 Layout ..... 671
54.14 Caption ..... 671
55 Grid Chart ..... 673
55.1 Grid Chart: Object Menu ..... 673
55.2 Dimensions ..... 675
55.3 Dimension Limits. ..... 675
55.4 Expressions ..... 675
55.5 Sort ..... 676
55.6 Style. ..... 676
55.7 Chart Properties: Presentation (Scatter- Grid Chart) ..... 676
55.8 Axes. ..... 680
55.9 Colors. ..... 680
55.10 Number ..... 680
55.11 Font ..... 680
55.12 Layout ..... 680
55.13 Caption. ..... 680
56 Pie Chart ..... 681
56.1 Pie Chart: Object Menu ..... 681
56.2 Dimensions. ..... 683
56.3 Dimension Limits. ..... 683
56.4 Expressions. ..... 683
56.5 Sort ..... 684
56.6 Style. ..... 684
56.7 Chart Properties: Presentation (Pie Chart) ..... 684
56.8 Colors ..... 685
56.9 Number ..... 685
56.10 Font ..... 685
56.11 Layout ..... 686
56.12 Caption. ..... 686
57 Funnel Chart ..... 687
57.1 Funnel Chart: Object Menu. ..... 687
57.2 General ..... 689
57.3 Dimensions ..... 689
57.4 Dimension Limits. ..... 689
57.5 Expressions ..... 690
57.6 Sort ..... 690
57.7 Style. ..... 690
57.8 Chart Properties: Presentation (Funnel Chart) ..... 690
57.9 Colors ..... 691
57.10 Number ..... 691
57.11 Font ..... 692
57.12 Layout ..... 692
57.13 Caption ..... 692
58 Block Chart ..... 693
58.1 Block Chart: Object Menu. ..... 693
58.2 General ..... 695
58.3 Dimensions ..... 695
58.4 Dimension Limits. ..... 695
58.5 Expressions ..... 696
58.6 Style. ..... 696
58.7 Chart Properties: Presentation (Block Chart). ..... 696
58.8 Colors. ..... 698
58.9 Number ..... 698
58.10 Font ..... 698
58.11 Layout ..... 698
58.12 Caption ..... 698
59 Gauge Chart ..... 699
59.1 Gauge Chart: Object Menu. ..... 699
59.2 General ..... 701
59.3 Dimensions ..... 701
59.4 Expressions ..... 701
59.5 Sort ..... 702
59.6 Style. ..... 702
59.7 Chart Properties: Presentation (Gauge Chart) ..... 702
59.8 Actions ..... 705
59.9 Colors. ..... 705
59.10 Number ..... 705
59.11 Font ..... 705
59.12 Layout ..... 706
59.13 Caption ..... 706
60 Mekko Chart ..... 707
60.1 Mekko Chart: Object Menu ..... 707
60.2 General ..... 709
60.3 Dimensions ..... 709
60.4 Dimension Limits. ..... 709
60.5 Expressions ..... 709
60.6 Sort ..... 709
60.7 Style. ..... 709
60.8 Chart Properties: Presentation (Bar- Line- Combo- Radar- Mekko Chart) ..... 710
60.9 Axes. ..... 712
60.10 Colors ..... 712
60.11 Number ..... 713
60.12 Font. ..... 713
60.13 Layout ..... 713
60.14 Caption. ..... 713
61 Pivot Table ..... 715
61.1 Using the Pivot Table. ..... 715
61.2 Pivot Table: Object Menu. ..... 718
61.3 General ..... 721
61.4 Dimensions ..... 721
61.5 Expressions ..... 721
61.6 Sort ..... 721
61.7 Chart Properties: Presentation (Pivot Table) ..... 721
61.8 Chart Properties: Visual Cues ..... 723
61.9 Chart Properties: Style ..... 724
61.10 Number ..... 725
61.11 Font ..... 725
61.12 Layout ..... 725
61.13 Caption. ..... 725
62 Straight Table ..... 727
62.1 Using the Straight Table ..... 727
62.2 Straight Table: Object Menu. ..... 727
62.3 General ..... 730
62.4 Dimensions ..... 730
62.5 Dimension Limits ..... 730
62.6 Expressions ..... 730
62.7 Chart Properties: Sort (Straight Table) ..... 731
62.8 Chart Properties: Presentation (Straight Table). ..... 732
62.9 Chart Properties: Visual Cues ..... 734
62.10 Chart Properties: Style. ..... 735
62.11 Number ..... 736
62.12 Font ..... 736
62.13 Layout ..... 736
62.14 Caption. ..... 736
63 Edit Expression Dialog ..... 737
63.1 File Menu. ..... 739
63.2 Edit Menu ..... 740
63.3 Settings Menu. ..... 740
63.4 Colormix Wizard. ..... 740
64 Boxplot Wizard. ..... 745
64.1 Boxplot Wizard Define Data ..... 745
64.2 Boxplot Wizard Presentation. ..... 746
65 Quick Chart Wizard ..... 747
65.1 Starting the Quick Chart Wizard ..... 747
66 Time Chart Wizard ..... 755
66.1 Starting the Time Chart Wizard. ..... 755
67 Statistics Chart Wizard ..... 761
67.1 Starting the Statistics Chart Wizard. ..... 761
67.2 Chi2-Test ..... 761
67.3 Paired T-Test ..... 762
67.4 Independent Samples T-Test ..... 763
68 Chart Expressions. ..... 765
68.1 Alternate States and Chart Expressions. ..... 766
68.2 Variable Text Expansion ..... 767
68.3 Operators ..... 767
68.4 Chart Aggregation functions ..... 770
68.5 Synthetic Dimension Functions. ..... 803
68.6 Other Functions. ..... 804
69 Examples ..... 831
69.1 Examples of Aggregate Qualifiers ..... 831
69.2 Examples of Alternate States in Chart Expressions ..... 832
70.1 Examples of Chart Ranking Functions. ..... 835
71.1 Examples of Chart Inter Record Functions ..... 839
72 Nested Aggregations and Related Issues ..... 845
73 Calculated Formulas ..... 851
73.1 Entering a Calculated Formula ..... 851
73.2 Error Messages. ..... 851
73.3 Calculated Formula ..... 851
74 Field Groups ..... 853
74.1 Hierarchic Groups (Drill-Down). ..... 853
74.2 Non-Hierarchic Groups (Cyclic). ..... 854
75 Custom Error Messages. ..... 855
Part 7 Number Formats ..... 857
76 Data Types in QlikView ..... 858
76.1 Data Representation inside QlikView. ..... 858
76.2 Number Interpretation ..... 858
76.3 Dates and Times. ..... 861
77 Format Codes for Interpretation and Formatting Functions ..... 865
77.1 Numbers ..... 865
77.2 Dates ..... 865
77.3 Times ..... 866
77.4 Time Stamps ..... 866
Part 8 Macros and Automation ..... 869
78 Using Automation and Macros with QlikView ..... 870
78.1 The QlikView Automation Interface ..... 870
78.2 How Automation and Macros Can Control QlikView. ..... 870
79 Internal Macro Interpreter ..... 873
79.1 Invoking Macros. ..... 873
79.2 Special Library Functions for JScript ..... 874
79.3 Edit Module ..... 875
79.4 Macro Debugger ..... 877
80 Getting Hold of a QlikView Document ..... 879
80.1 Accessing QlikView Documents from the Outside ..... 879
80.2 Accessing QV Documents from the Internal Macro Interpreter. ..... 879
81 VBScript Function Calls from Script ..... 881
81.1 Transfer of Parameters ..... 882
82 Using Macros in QV Documents on the QV-Server ..... 883
82.1 Macros on QlikView Server ..... 883
82.2 Macro functionality limitations. ..... 883
82.3 Macro trigger limitations ..... 883
82.4 VBScript functions ..... 884
82.5 Server Side Export ..... 884
Part 9 Appendix ..... 885
A Limitations and Requirements ..... 886
A. 1 Limitations of the Amount of Data to be Loaded. ..... 886
A. 2 System Requirements ..... 886
A. 3 Dynamic Link Libraries (dll's) Needed. ..... 887
B Keyboard Command Shortcuts. ..... 889
B. 1 File Menu Command Shortcuts. ..... 889
B. 2 Edit Menu Command Shortcuts. ..... 889
B. 3 View Menu Command Shortcuts ..... 890
B. 4 Selections Menu Commands Shortcuts ..... 890
B. 5 Settings Menu Command Shortcuts. ..... 890
B. 6 Bookmarks Menu Command Shortcuts. ..... 890
B. 7 Tools Menu Command Shortcuts. ..... 890
B. 8 Object Menu Command Shortcuts (List box, Statistics box and Open Multi Box) ..... 891
B. 9 Script Keyboard Shortcuts. ..... 891
B. 10 F Key Keyboard Shortcuts. ..... 891
C Application Performance Optimization ..... 893
C. 1 Introduction ..... 893
C. 2 Count (Distinct 'FieldName') ..... 893
C. 3 If ( Condition(Text),....). ..... 894
C. 4 Sum ( If (Condition, ‘FieldName’...)) ..... 894
C. 5 If ( Condition, Sum('FieldName')..), ..... 895
C. 6 If ( Condition1, Sum('FieldName'), If (Condition2, Sum('FieldName') ..... 895
C. 7 Sorting Text ..... 896
C. 8 Dynamic Captions and Text Objects. ..... 896
C. 9 Macro Triggers ("on change"). ..... 896
D Frequently Asked Questions ..... 897
D. 1 Installation. ..... 897
D. 2 QlikView Documents. ..... 897
D. 3 Scripts and Loading Data ..... 898
D. 4 QlikView Logic ..... 899
D. 5 Layout ..... 900
D. 6 Sharing QlikView Documents with Other People. ..... 901
E Data Protection Issues ..... 903
E. 1 Data Security and Integrity ..... 903
E. 2 The Right to Alter Data ..... 903
E. 3 The Right to View Data ..... 903
E. 4 Data Integrity. ..... 903
F Backus-Naur Formalism ..... 905
Glossary ..... 907
Index ..... 921

## 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. $1,2,3, \ldots$ ) indicate procedures with two or more sequential steps.
- Bulleted lists, such as this one, provide information, and do not indicate procedural steps.


## Mouse Conventions

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


## Keyboard Conventions

- Key names appear in capital letters, e.g. "Press ENTER".
- The return key and the enter key perform the same action in QlikView.
- A plus sign " + " used between two key names indicates that you must press both keys at the same time. E.g., ctrl+s means that you should press the ctrl-key while pressing s.
- A comma sign "," used between two key names indicates that you must press the keys sequentially.


### 1.2 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 home page (see above).

## 2 What's New in QlikView 11?

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 with several 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 QlikView Desktop

## General Features

## Improved Hierarchy Loads

The hierarchy loads have been redesigned. The hierarchy prefix executes in half the time compared to QlikView 10. Other changes include improved scalability with very large hierarchies. There has been no change to the syntax for hierarchy loads.

## Drop-Down Selection Tool for Tabs in the Load Script

A drop-down selector for the script tabs has been added to the Script Editor toolbar. This provides users with an alternative to clicking on load script tabs to navigate through the tabs. For more information see Edit Script Dialog (page 173).

## OEM Island

QlikView provides a new mechanism that can restrict QlikView documents, that are tagged with an application ID, from running on any other QlikView Server without a matching ID. This feature is reserved for use by OEM partners only.

## BNF Notation for Load Script Tool Tips

The QlikView script editor now displays BNF notation (syntax) in the tool tips.

## Document Extensions

Document Extensions are Extension Objects that contain no user interface. They can be used to customize the generated AJAX pages or used as sources of data for integration purposes. Although this feature was scheduled for this release, it is only available from a separate build (20747).

## Context Sensitive Help for the Script Editor

Context sensitive help was broken in QlikView 10. It has been fixed for QlikView 11. The help text, however, has not been completely updated for QlikView 11 at this time.

## Alternate States

QlikView allows users to detach charts from the underlying data model. When a chart is detached the object no longer responds to user selections. When the user re-attaches the object, it is instantly updated to reflect the current state of the selections. Alternate States are similar to this. 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. Alternate States are not accessible in the load script. They are a UI feature.

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. See also Alternate States (page 122)

## Session Collaboration

QlikView 11 adds two new collaboration features; the first is called Session Collaboration. This feature allows server based QlikView users to share their QlikView session with other people, inside or outside of their organization. With this feature, users can discover and develop new analyses in a collaborative environment. Session Collaboration is only available when connected to a QlikView Server and using the AJAX client. The QlikView Plugin client does not support this feature.

## Notes and Comments

The second new collaboration feature is Notes and Comments. This feature allows users to create and share notes about QlikView objects. Notes are shared with other QlikView users and these others users can respond by leaving their own comments. The users have the ability to save a snapshot (bookmark) of the data with each Note and Comment.

## Dimension Limits

QlikView has had the capability to present "Others" and "Totals" values in various chart types. QlikView also allows users to configure charts to display the top or bottom $X$ values in several charts. All of this manipulation was previously managed by the charting engine. Starting with QlikView 11, these capabilities have been moved to the calculation engine. This means users will be able to set these options across all chart types, except for Gauge Charts and Pivot Tables. There is a new tab in the chart property dialog called Dimension Limits that allows users to configure these options.

## Conditional Enablement of Expressions and Dimensions

It is now possible to conditionally enable and disable dimensions and expressions. There are new checkboxes and expression fields on the Dimensions and Expressions tabs in the property dialogs for use with this feature.

## Source Control Integration

QlikView 11 has the capability to integrate with Source Control systems. The feature is available when using QlikView Desktop. There is a new menu option in the File menu that allows QlikView Developers to connect a QlikView document to a source control system. This feature works with the XML project files that QlikView Desktop produces and offers basic integration into third party source control systems. The source control system must have an implementation of the MSSCCI API in order to use this feature. QlikView 11 Source Control integration is compatible with Microsoft Team Foundation Server.

## Mapping Loads

You can now perform mapping loads directly from a QVD file. There is no longer a need to un-optimize the QVD load while performing a mapping load. This has been available since QlikView 10 but was not documented.

## Container Object

The container object has a new way of displaying objects, grid mode. This allows users to view and interact with more than one object at a time in a container object.

### 2.2 AJAX Client Improvements

Apart from the new layout features listed above, a number of AJAX specific features have been added:

## Pivoting in Pivot Tables

The goal is to have parity between the C client and the AJAX client. It is now possible to pivot a pivot table in the AJAX client. For example, if a pivot table has 3 dimensions and several expressions, it is now possible to arrange the columns in any order.

## UI Upgrades

A number of minor upgrades have been added to the AJAX client:

- Order by columns in list boxes.
- A new tab row is available in IE9, Firefox 4 (and higher) and Chrome 12 (and higher).


## Property Dialogs

The property dialogs have a new look and feel in the AJAX client. These changes were made to improve the look and usability of the dialogs. The new dialogs are available in the AJAX client and in WebView on QlikView Desktop.

### 2.3 Server Components

## QMC Help

Context sensitive help has been added to the QMC.

## LDAPs

The configurable LDAP DSP (Directory Service Provider) for LDAPs (LDAP via SSL) support has been enhanced to better support our customer needs.

## Audit Logging by Document

Some organizations require that a user audit log is generated, so that every query is registered and it is possible to look at "who did what" retroactively. QlikView 10 Server can do this; however the switch is for the entire server - either all documents are logged, or no document is logged. In Version 11 this logging can now be set per individual document.

## Enable and Disable Document Download per Document \& User

In many situations the system manager will want to prevent any "hard" data coming out of the QlikView Server. In these situations downloads of the .qvw file, printing and exporting are not allowed; only the interactive session with the QlikView Server is permitted. This functionality is available in Version 10 at the document level. In Version 11 it is now possible on a per user level as well.

## Enable and Disable Export and Print per Document \& User

In many situations the system manager will want to prevent export to excel and print. In Version 11 it is now possible on a per document and user level.

## Scriptlets for QVD creation

It is now possible to add creation of QVD files via a scriptlet. This is done under "Supporting Tasks".

## Distribution to E-mail within a QVW Document

It is now possible to distribute a QVW file to the e-mail recipients defined in a field in the document.

## Alert E-mail to Document Administrators

It is now possible to send alert e-mail to document administrators.

## License Tracking

The use of licenses is added to the QVS logs.

## Distribution and Reload Performance

The performance of reload and distribution has been improved:

- QlikView 11 Publisher can now open source documents without data prior to executing a reload task. There is no need to load the document data into memory and then perform a reload of the document.
- Loop and reduce performance will be improved in a future release.


## Management Console UI

- UI inconsistencies have been cleaned up.
- The performance of refreshing tables in the QMC has been improved overall.
- The Status page is drawn and updated faster.
- It is now possible to expand all of a task chain by right-clicking.
- It is now possible to remove a user from all distributions where the user is explicitly targeted.
- Search and filter is added to Source and User Documents.
- Alerts are added for service status.
- Clustering and user types is made more consistent among the services.


## Reduction with Lock Fields

In previous versions of QlikView a reduction was affected by lock fields (by in effect reduce on the locked values). In version 11 the reduction will ignore any locked field.

## Logging in Distribution Service

Logging and error handling is not completely reliable and sometimes confusing. One real error could in some cases show as three or more errors. A task that finished OK can sometimes have many warnings because the QVB has not responded for a few seconds. This behavior is now fixed.

## QMC and QMEC merged into QMC

The QMC is removed and QEMC is renamed to QMC.

## Access Point Remake

The Access Point has been given a new look and feel, including some new search capabilities.

## EDX Enhancements

Starting an EDX will now return a session id to allow interrogation of the status of the session rather than on the task. When the session is done the status will contain a list of all the tasks (and session id's) that have been triggered, allowing for continuous tracking of the status. This functionality is available through the API.

## Load-Balancing Improvements

The reporting of memory and CPU loads on the physical QlikView Servers has been improved. The load balancing algorithm has also been improved. It is possible to change the actual load balancing formula used, but only through the configuration file. This is intentional and not in the UI as it should not be tampered with in a normal installation.

## MSI - Installation for QlikView Server

The MSI has been given a usability overhaul.

### 2.4 WebParts and WorkBench

The QlikView Workbench has been updated with a number of new features:

- The QlikView Workbench now includes a wizard for building Extension Objects.
- Workbench could previously only be used within an ASP.NET web site project. It is now possible to connect to a QVW via JavaScript / div tag without the need of a .NET dll component.


### 2.5 APIs

Version 11 will include updates to all APIs, including COM, QlikView Management Service and JavaScript. All new APIs are designed with new functionality and backward compatibility in mind. In addition, QlikView will expose a new back-end API to manage access to QlikView Server clusters and retrieve document lists.

## COM API

QlikView provides new APIs for the automation of Alternate States.

## QlikView Management Service API

This release contains the following updates to the QlikView Management Service API:

- Service status
- QlikView Server and QlikView Publisher license configuration and management (not CALs)
- QlikView Distribution Service settings
- User document collaboration configuration (for restricted mode)
- Opening/accessing source document data (e.g. fields and bookmarks)
- QVS statistics
- DSC user lookup and validation
- Importing tasks/remote QMS communication


## QlikView JavaScript API

The version 11 JavaScript Documentation update is scheduled for the RC release.

## 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 purchasing QlikView an e-mail is received with the information about the license agreement and with the serial number(s) and the control number(s) needed.
QlikView is designed to run on an IBM PC compatible computer with Windows XP SP3, Windows Vista, Windows 7 Windows Server 2003 or Windows Server 2008. For detailed information on the system requirements, see Appendix (page 885).

### 5.1 Starting the Installation Program

Do the following:

1. Go to www.qlikview.com/download.
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.

### 5.2 When the Installation Program Is Running

1. The first dialog that opens is a "Welcome" dialog, just click Next.
2. In the second dialog that opens, choose region and click OK.
3. The software license agreement opens. Read it, and click "I accept the license agreement" (if this is the case), then click Next.
4. In the User Information dialog you can specify for whom the installation will be personalized. Click Next to continue.
5. The installation type dialog appears. If Complete is chosen, the following features will be installed to the default path, C:\Program Files\QlikView: The QlikView program and its Help file, the QlikView Internet Explorer Plug-In, Examples of Custom objects and examples of QlikView functionality and features. If Custom is chosen, a dialog where you can specify which features to install and where to install them will appear when you click Next. Make the selections and click Next.
6. The installation is now ready to start. Click Next to start it.
7. 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 QlikViewx 64 .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 and Windows 7. On older systems the file is found in $C: \backslash$ Documents and Settings \username\Application Data\QlikTech \QlikView.
The settings for QlikView OCX are found in the same file.

### 5.3 Serialization

If you get your QlikView license from a QlikView Server or if you have a Personal Edition of QlikView no serialization is necessary. It is however possible to acquire a serial number for this product and enter it in User Preferences: License 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.

### 5.4 Network Distribution of QlikView

This type of installation is useful if you have many licenses and want to make sure that all users run the same version of QlikView. Each user still needs a separate license number.
To distribute a QlikView installation, you need the installation DVD. Serial numbers and control numbers are not necessary until you start your first QlikView session on the client computer.
QlikView 11 uses the Microsoft Windows Installer technique (MSI packaging) and a specific server installation program is no longer needed. The Microsoft Windows Installer technique has built-in support for this type of installations.
To find out more about the ways to install an MSI-package please visit the Microsoft homepage for Windows installer:

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


### 5.5 License Enabler File Editor



## 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.

### 5.6 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 $\mathrm{c}: \backslash \mathrm{windows} \backslash$ SysWOW64. According to Microsoft, in the 64-bit versions of the operating systems, the system 32 folder, usually c:lwindows system 32 , 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 64bit 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 decition 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 $\mathbf{O K}$ 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

## 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 905)):

## [ 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.

## /I

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 39).

## /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 $/ \mathbf{r} / v$ Myvar $=123 \mathrm{abc} . q v w$

## 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 278))

### 7.2 The Start Page



## The Getting Started page of the QlikView start page

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

Getting started Contains information and links that can help explore the wealth of possibilities in QlikView. This includes direct links to downloading the Tutorial and to the Getting Started Wizard as well as links to selected resources at www.qliktech.com.

| Recently Opened | Contains a list of recently opened documents and web pages. Just click on one |
| :--- | :--- |
| Documents | 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 38)) 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. |  |
| Contains a list of the documents and web pages previously added to the favor- |  |
| ites list. Unlike the list of recently used documents the favorites list remains |  |

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

| Connect to Server |  |  |  |  | $x^{2}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Server |  |  |  |  |  |
| qvp://selun-cen | $\checkmark$ | Connect | Clear List |  |  |
| $\begin{aligned} & 8 \text { selun-cen } \\ & \square \text { mobile } \\ & \square \mathrm{QW} \end{aligned}$ | AuthenticationUse NT IdentityAnonymousAlternate Identity |  |  |  |  |
| Name | Time |  | Size | Users | - |
| (1llObjects-Eng.qvw | 2010-0 | -05 18:20:22 | 139.3 KB | 0 |  |
| Data Visualization.qvw | 2010-0 | -02 10:03:44 | 7.7 MB | 0 |  |
| - Films.qvw | 2010-0 | -02 10:03:36 | 341.8 KB | 0 | 三 |
| - Finance Controlling.qvw | 2010-0 | -02 10:03:36 | 300.8 KB | 0 |  |
| O Online Sales.qvw | 2010-0 | -02 10:03:44 | 648.0 KB | 0 |  |
| (1) Presidents.qVw | 2010-0 | .02 10:03:38 | 578.8 KB | 0 |  |
| - Presidents_AJAX.qvw | 2009-1 | -17 14:46:03 | 577.8 KB | 0 |  |
| Whats New in Qv10.qvw | 2010-0 | 16 10:02:16 | 19.2 MB | 0 | - |
|  | Open | Can | cel | Help |  |

## 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 42) 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.

[^0]| AuthenticationSelect how to identify yourself to the QlikView Server. <br> The following three authentication methods are available: <br> Use NT Identity <br> Uses authenticated NT identity. This only works if the computer is connected <br> to the same Windows NT domain as the QlikView Server. <br> Anonymous <br> Uses anonymous login. It is only possible to connect if the QlikView Server is <br> set to allow anonymous users and it is only possible to see documents with <br> file permissions for the QlikView Server anonymous account. The connection <br> pseudo-URL under Server will be complemented by a @ before the server <br> name. <br> Alternate Identity <br>  <br> Uses a specific NT user name known on the QlikView Server's domain. The <br> user name is typed in the User ID edit box. It will prompt for the cor- <br> responding user password when pressing the Connect button. The connection <br> pseudo-URL under Server will be complemented by username@ before the <br> server name. <br> Flat Mode$\quad$With this setting enabled, the bottom pane of the dialog will display the con- <br> tents 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.
@ withoutusername 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 38).

## Scheduling Programs

Windows has a Task Scheduler service. The services are controlled from the Computer management applet in the Control Panel (Windows Vista) or directly from the Control Panel (Windows XP). Make sure that Status is set to Started (and Startup to Automatic for the service to automatically start after a reboot) for the relevant service.
The easiest way to submit a batch job is usually to create a batch file with the relevant commands. The file, named e.g. MyBatch. cmd, could e.g. contain the following commands:
C:\qlikview\qv.exe /r C:\qlikview\example\file.qvw

## The Schedule Service

The schedule service normally runs under the system account and thus has some limitations. If 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 401).

## ODBC

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

Control Panel.

## 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 Data Source

Step 2 - Verify Data Presentation
Step 3 - Save File
Step 4 - Select Chart Type
Step 5 - Define Dimensions and Add List Boxes
Step 6 - Add Expression
page 45
page 46
page 47
page 47
page 48
page 49

## 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 Data Source



Step 1 of the Getting Started wizard
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 Next to continue.

## Step 2 - Verify Data Presentation



## Step 2 of the Getting Started wizard

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 to continue.

## Step 3 - Save File

| © Save As |  |  |  | $x^{2}$ |
| :---: | :---: | :---: | :---: | :---: |
| Save in: | ]. V10 |  | (9) 企 |  |
| Recent Places <br> Desktop <br> Libraries <br> Computer <br> Network | Name |  | Date modified | Type |
|  | ]3illObjectsEN-prj |  | 2010-05-05 16:08 | File folder |
|  | 13. ProjectFile-prj |  | 2010-05-04 13:35 | File folder |
|  | AjaxTest.quw |  | 2010-05-04 08:10 | QlikView [ |
|  | (1llObjectsEn.quw |  | 2010-05-05 16:34 | QlikView [ |
|  | ( Container.quw |  | 2010-05-04 16:20 | QlikView [ |
|  | FieldTags.qvw |  | 2010-03-24 14:30 | QlikView I |
|  | - Mekko.qvw |  | 2010-04-19 16:54 | QlikView [ |
|  | ProjectFile.qvw |  | 2010-05-04 13:35 | QlikView [ |
|  | Q QDS.qvw |  | 2010-05-03 17:14 | QlikView [ |
|  | SingleSheetObject.quw |  | 2010-05-05 16:48 | QlikView I |
|  | WebView.quw |  | 2010-04-27 08:36 | QlikView [ |
|  | III |  | $\square$ | + |
|  | File name: |  | $\checkmark$ | Save |
|  | Save as type: | QlikView Files (qvw) | $\checkmark$ | Cancel |

The Save As dialog of the Getting Started wizard
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 to continue and create a chart, or click Finish to save the document and close the wizard.
Step 4 - Select Chart Type


Step 4 of the Getting Started wizard

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 to continue.

## Note!

In this step the Back button is disabled.

## Step 5 - Define Dimensions and Add List Boxes



Step 5 of the Getting Started wizard
Select one or more dimensions in the drop-down boxes.
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.
This step is automatically skipped for gauge charts because they do not normally have any dimensions.
After finishing this wizard it is possible to add more dimensions or change dimensions at any time via the Chart Properties dialog.
To display the values of the defined dimensions in list boxes as well as the chart, leave the check box Show List Boxes checked.

Click Next to continue.

## Step 6 - Add Expression



Step 6 of the Getting Started wizard
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.
After finishing this wizard it is possible to add more expressions or change expressions at any time via the Chart Properties dialog. The chart properties dialog is reached by right-clicking on the chart and selecting Properties....
To get information on where to go next, leave Launch help to read about ways to develop your QlikView document checked.
Click Finish to close the wizard and view the chart.
$\longrightarrow$

## 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
- $S B<i d>. x m l$
- 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 Microsoft Source Control Plug-in API (MSSCCI) is the provider used for this function. There are many MSSCCI provider DLLs that enable integration with different source control systems and are available for downloading. The QlikView implementation is based on the 1.2 version of the MSSCCI API.
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. The file is named [QlikViewDocumentName]prj.scc 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 MSSCCI 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. 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, then QlikView will try to work "offline", 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 change sets. As a result, rollback between randomly selected change sets does not guarantee that the document will be in a stable state.

## 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 72) 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:

| New | Opens a new QlikView window and allows to create a new QlikView File <br> (page 51). |
| :--- | :--- |
| Open... | Opens a new QlikView window and allows to open a QlikView file or a <br> Table file. Opening a table file automatically opens the File Wizard: Type <br> (page 196). This command can also be invoked by the following keyboard <br> shortcut: CTRL+O. |
| Open in Server... | Opens the Connect to Server (page 41). In this dialog it is possible to con- <br> nect to a QlikView Server and browse for a document to open in client <br> mode. |
| This command can also be invoked by the following keyboard shortcut: <br> CTRL+SHIFT+O. |  |
| Note! <br> When opening documents as a client to QlikView Server the possibilities are restricted <br> on what is possible to do with the document compared to when opening local doc- <br> uments. As a client it is not possible to add or delete sheet objects, depending on whether <br> the document supports collaboration. Sheets cannot be added or deleted. Furthermore, <br> not all properties dialogs can be accessed, macros changed, the script accessed, data <br> reloaded or the document saved. |  |
| This command is only available with documents opened on QlikView |  |
| Refresh Doc- $\quad$Server and when there is a newer version of the document available on the <br> server. When invoking a refresh access is gained to the latest data, while |  |
| umentmaintaining the session including selections and layout state. |  |
| Opens the Open URL dialog. In this dialog type a valid URL to any web |  |


| Save | Saves the present configuration in a QlikView file. Data, script and layout <br> are saved. This command can also be invoked by the following keyboard <br> shortcut: CTRL+S. |
| :--- | :--- |
| Save As... | Saves the present configuration in a new QlikView file under a new name. <br> This command can also be invoked by the following keyboard shortcut: <br> F12. |
| Save Link... | Saves a link to a document opened on a QlikView Server as a text file on <br> the local machine. The file will have the qvw extension but will not con- <br> tain any data or layout. When opening such a link document QlikView will <br> attempt to reconnect to the server and open the document on the QlikView |
| Server. This command is not available for local documents. |  |

## 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 52).

## 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 checkout 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.

| Export | Export Contents... <br> This alternative is only available when an exportable sheet object is active. It then performs the export operation found on the sheet object's Object menu. |
| :---: | :---: |
|  | Export Sheet Image... <br> Opens a dialog for saving an image of the current sheet to file. |
|  | Export Document Layout <br> Opens a dialog for saving the document layout as an XML file. No data from the document are saved in the XML file. |
|  | Export Sheet Layout... <br> Opens a dialog for saving the current sheet's layout as an XML file. No data from the document are saved in the XML file. |
| Import | Import Document Layout <br> Opens a dialog for importing the document layout of another document saved as XML. No data are imported. |
|  | Import Sheet Layout <br> Opens a dialog for importing the layout from another sheet saved as XML. No data are imported. |
| Edit Script... | Opens the Edit Script Dialog (page 173). In this it is possible to write and execute scripts that open and connect to databases. |
| Reload | Executes the current load script and reloads data to the active QlikView document. This needs to be done if the contents of the database have been changed since the last time a reload was made. This command can also be invoked by the following keyboard shortcut: CTRL+R. |
| Partial Reload | Executes the current load script, including all script commands, such as Drop Table (page 232), and reloads data to the active QlikView document. However, only those tables whose load and select statements are preceded by the Replace (page 260) or Add (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. |
| Reduce Data | Opens a menu containing the following two commands: |
|  | Keep Possible Values <br> Reduces the QlikView database by removing all excluded values. |
|  | Remove All Values <br> Creates a template by removing all the values from the QlikView database, but keeping the database structure and the layout. |
| Table Viewer... | Opens the Table Viewer (page 181) 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. |

Recent documents A listing of the last QlikView documents used．The number of documents list shown can be set in the Settings：User Preferences dialog．Default number is 8 ．Selecting one of these is equivalent to the Open 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：

| Undo Layout Change | Undoes the latest layout change，including moving，sizing and remov－ <br> ing of sheet objects as well as changes to sheet object properties． <br> Removing sheets，editing the sheet properties or document properties <br> can also be undone．This command can also be invoked by the fol－ <br> lowing keyboard shortcut：CTRL＋Z． |
| :--- | :--- |
| Redo Layout Change | Redoes the latest undone layout action．This command can also be <br> invoked by the following keyboard shortcut：CTRL＋Y． |
| Cut | Moves the selected sheet object（s）to Clipboard for pasting elsewhere <br> in the QlikView document．If only one sheet object is active a bitmap <br> image of the object，which can be pasted into other programs，will also <br> be put into Clipboard．This command can also be invoked by the fol－ <br> lowing keyboard shortcut：CTRL＋X． |
| Copy | Copies the selected sheet object（s）to Clipboard for pasting elsewhere <br> in the QlikView document．If only one sheet object is active a bitmap <br> image of the object，which can be pasted into other programs，will also <br> be put into Clipboard．This command can also be invoked by the fol－ <br> lowing keyboard shortcut：CTRL＋C． |
| Faste | Pastes one or several sheet objects from Clipboard back to the Qlik－ <br> View document．This command can also be invoked by the following <br> keyboard shortcut：CTRL＋V． |
| Femave Painter | The Format Painter（page 7l）tool makes it possible to copy for－ |
| matting from one sheet object to another． |  | a searchable sheet object is active．

Advanced Search

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 104) 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:

| Sheets | Upon selection, a cascade menu displays a list of all sheets used in the <br> document, sorted as they appear from left to right. |
| :--- | :--- |
| Toolbars | Upon selection a cascade menu, where it is possible to enable any tool- <br> bar to be displayed with the document, opens. The following toolbars <br> can all be employed independently of each other: Standard Toolbar, <br> Navigation Toolbar, Design Toolbar, Sheets Toolbar and Book- <br> marks Toolbar. The last entry in the list is the Customize (Tool- <br> bars)... (page 72) option allows the configuration of the toolbar <br> according to the users own preferences. <br> The Standard, Navigation and Design toolbars have each been pre- <br> defined with a suitable selection of menu commands, as hinted by the <br> names. The Sheet and Bookmark toolbars are actually defined as drop- <br> down lists, offering an optional means for navigation between sheets <br> and bookmarks, respectively. Note that the contents and functionality <br> of any toolbar can be configured in a number of ways via the Cus- <br> tomize option. <br> Tip! 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. |  |


| Design Grid | Toggles the design grid, sheet object placeholders for active object(s) <br> and snap-to-grid for sizing and moving objects in the layout on or off <br> This command can also be invoked by the following keyboard short- <br> cut: CTRL+G. |
| :--- | :--- |
| Turn on/off WebView |  |
| mode | Toggles WebView mode, which uses the internal web browser in Qlik- <br> View to display the document in AJAX mode, on and off. |
| Current Selections... | Opens the Current Selections (page 111) dialog where selections are <br> listed by field name and field value. This command can also be <br> 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:

| Back | Reverts to the preceding logical state. Applies to value selections and <br> all the commands in the Selections menu. QlikView maintains a list of <br> the 100 latest states. Each Back will take one step further back in the <br> list. This command can also be invoked by the following keyboard <br> shortcut: SHIFT+LEFTARROW. |
| :--- | :--- |
| Forward | Reverts to the logical state before a Back command. Toggle between <br> two states by alternating between the Back and Forward commands. <br> This command can also be invoked by the following keyboard short- <br> cut: SHIFT+RIGHTARROW. |
| Lock | Locks all the current value selections. This command can also be <br> invoked by the following keyboard shortcut: CTRL+SHIFT+L. |
| Cnlock | Unlocks all the currently locked value selections. This command can <br> also be invoked by the following keyboard shortcut: CTRL+SHIFT+U. |
| Clear Applies the start selection of a QlikView document, which can be con- |  |
| figured, see Set Clear State below. |  |
| This command can also be invoked by the following keyboard short- |  |
| cut: CTRL+SHIFT+D. |  |

### 10.5 Layout Menu

The Layout menu is a drop-down menu at the top of the screen, and contains the following commands:
Add Sheet...
Adds a tabbed sheet in which a new set of sheet objects can be dis-
played.
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.

| Promote Sheet | Moves the (tab of the) current sheet one step to the left (forwards). |
| :--- | :--- |
| Demote Sheet | Moves the (tab of the) current sheet one step to the right (back). |
| Remove Sheet | Deletes the active sheet and everything on it. <br> Select Fields... |
| Opens the Fields property page of the Sheet Properties sheet. One or <br> several field names can be selected from a list. The selected fields will <br> be displayed in default configured list boxes on the active sheet. To set <br> the specific properties of a list box right click on it and select Prop- <br> erties from the float menu (or from Object in the Main Menu). This <br> command is not available when working with documents on QlikView |  |
| Server. |  |

### 10.6 Settings Menu

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

| User Preferences... | Opens the User Preferences (page 77) dialog that contains settings <br> that the user does not normally change when switching to another doc- <br> ument. This command can also be invoked by the following keyboard <br> shortcut: CTRL+ALT+U. |
| :--- | :--- |
| Document Properties... | Opens the Document Properties: General (page 408) dialog that con- <br> tains settings for the entire document. Some common properties for the <br> sheet objects in the document can also be set in this dialog. This com- <br> mand can also be invoked by the following keyboard shortcut: <br> CTRL+ALT+D. |
| Sheet Properties... | Opens the Sheet Properties: General (page 447) dialog that contains <br> settings for the current sheet. Some common properties for the sheet <br> objects on the sheet can also be set in this dialog. This command can <br> also be invoked by the following keyboard shortcut: CTRL+ALT+S. <br> See the links at the bottom of this page. |
| Variable Overview... | Opens the Variable Overview (page 161) dialog where all non-hidden <br> variables and their values are shown in a single list. This command can <br> also be invoked by the following keyboard shortcut: CTRL+ALT+V. |
| Expression Overview... | Opens the Expression Overview (page 163) dialog where all doc- <br> ument, sheet and sheet object expressions can be shown and centrally <br> maintained in a single list. This command can also be invoked by the <br> 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:
Document Bookmarks The first ten document bookmarks of the active document can be retrieved from this list.

User Bookmarks
The first ten personal bookmarks linked to the active document can be retrieved from this list.

| Add Bookmark | Opens the Add Bookmark Dialog (page 125) dialog, where the book- <br> mark name can be edited. This command can also be invoked by the <br> following keyboard shortcut: CTRL+B. |
| :--- | :--- |
| Replace Bookmark $\quad$The first ten document bookmarks are listed above the first ten personal <br> bookmarks of the active document. The command replaces the selec- <br> tion state of the selected bookmark with the current state. |  |
| Remove Bookmark $\quad$The first ten document bookmarks are listed above the first ten personal <br> bookmarks of the active document. The command removes the selected <br> bookmark. |  |


| More... | Opens the Bookmarks dialog, where all previously created bookmarks <br> for the document can be retrieved. |
| :--- | :--- |
| Import... | After browsing for and selecting a previously saved bookmark (.qbm) <br> file, the Import Bookmarks (page 129) dialog will open to make it pos- <br> sible to import bookmarks. |
| Export... | Opens the Export Bookmark(s) (page 129) dialog where selected book- <br> marks 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 134) 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 134).

### 10.9 Tools Menu

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

Open QlikView Acces-
sPoint

Open QlikView Management Console

## Quick Chart Wizard... Opens the Quick Chart Wizard (page 747), which helps creating a simple chart in a quick and easy way, without bothering about the great number of different settings and options available. <br> Time Chart Wizard... <br> Statistics Chart Wizard... <br> Box Plot Wizard

Alerts... $\begin{aligned} & \text { Opens the Alerts (page 150) dialog where alerts can be defined and } \\ & \text { edited. }\end{aligned}$

| Alert Wizard... | The Alert Wizard (page 154) helps with the task of defining an alert. |
| :--- | :--- |
| Theme Maker Wizard | Invokes the Theme Maker Wizard (page 583) for creating a new theme <br> or editing an existing theme. <br> More details about themes can be found under Layout Themes (page |
| 581). |  |

### 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:
Cascade Will arrange windows so that they overlap.
Tile Will arrange windows as non-overlapping tiles.
Arrange Icons
Will arrange icons at the bottom of the window.
Available Documents
Opens the Available Documents (page 63) 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:

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

| Contents | Starts the QlikView help. |
| :--- | :--- |
| Using Help | Gives some useful advice on how to use the QlikView help. |
| Show Start Page | Shows the start page each time QlikView starts. If the start page should not <br> appear when the program starts, deselect the Show Start Page When Launch- <br> ing QlikView check box. |
| QlikTech on the Web | Opens up further options that redirects to the QlikTech home page. <br> Opens the QlikView Update dialog from which it is possible to make Qlik- <br> Qiew contact the QlikView update server to see if any program updates are <br> available. |
| License Update... | Opens the License Enabler File Editor dialog where the user can review or <br> edit the License Enabler File. |
| Document Support | Opens the Document Support Information (page 65) dialog, displaying a list <br> of support data pertaining to the active document. |
| Info... | Opens the About dialog which displays the QlikView version, serial number |
| About QlikView... | 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.

| New File | Opens a new QlikView window and allows to create a new QlikView <br> File (page 51). This command can also be invoked by the following key- <br> board shortcut: CTRL+N. |
| :--- | :--- |
| Open File | Allows the opening of a QlikView File (page 51) or Table file in a new <br> QlikView window. Opening a table file automatically opens the File <br> Wizard: Type (page 196). This command can also be invoked by the fol- <br> lowing keyboard shortcut: CTRL+O. |
| Refresh $\quad$This command is only available with documents opened on QlikView <br> Server and when there is a newer version of the document available on <br> the server. When invoking a refresh access is gained to the latest data, <br> while maintaining the session including selections and layout state. |  |
| SaveSaves the active document as a file. The default file format is set in User <br> Preferences: Save (page 81). |  |
| Clicking the print tool immediately effects the printout of the selected <br> sheet object according to default printer settings, bypassing the Print: <br> General (page 99) property page. The tool will be grayed if no printable <br> object is selected. This command can also be invoked by the following <br> keyboard shortcut: CTRL+P. |  |


| Print as PDF | Opens the Print：General（page 99）with the QlikViewPDF printer pre－ selected． <br> After clicking the Print button you will be prompted for a file name for the PDF output file．This button is only available if a PDF printer is avail－ able on the system．The QlikViewPDF printer must be downloaded from QlikView＇s homepage and installed separately． | 嗗 |
| :---: | :---: | :---: |
| Edit Script | Opens the Edit Script Dialog（page 173）dialog，making it possible to write and execute Script that open databases and retrieve data into Qlik－ View．This command can also be invoked by the following keyboard shortcut：CTRL＋E． | 近 |
| Reload | Re－executes the current Script，updating the associative QlikView data－ base 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． | 景 |
| Undo Layout Change | 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． <br> QlikView maintains a list of the latest layout changes．Each Undo Lay－ out Change command makes it take one step backward in the list．Cer－ tain operations，e．g．Reload and Reduce Data will empty the Undo／Redo buffer．This command can also be invoked by the following keyboard shortcut：CTRL + Z． | 4 |
| Redo Layout Change | Redoes the latest undone layout action．Each Redo Layout Change com－ mand makes it take one step forward in the list，as long as there are undone actions to redo． <br> Certain operations，e．g．Reload and Reduce Data will empty the Undo／Redo buffer．This command can also be invoked by the following keyboard shortcut：CTRL＋Y． | $\cdots$ |
| Search | Opens the Search（page 113）box for the active object．This command can also be invoked by the following keyboard shortcut：CTRL＋F． | 樟 |
| Current Selections | Opens the Current Selections dialog in which it is possible to see the selections that are active．This command can also be invoked by the fol－ lowing keyboard shortcut：CTRL＋O． | $\checkmark$ |
| Quick Chart Wizard | Opens the Quick Chart Wizard（page 747）it allows to create a chart in a quick and easy way，without bothering about the great number of dif－ ferent settings and options available． | 抒 |
| Add Bookmark | Opens the Add Bookmark Dialog（page 125）dialog，where the book－ mark name can be edited．This command can also be invoked by the fol－ lowing keyboard shortcut：CTRL＋SHIFT＋B． | $\underline{1}$ |
| Help Topics | Opens the QlikView help． | （？） |
| Context Help | Displays specific help concerning the chosen object，e．g．a menu com－ mand．After clicking this button，simply move the question mark to the object on which help is needed． | ？ |

### 11.3 Navigation Toolbar

```
\vdots\mathbb{| Clear - | (5) Back (5) Forward| ( B2 Lock Ns)Unlock =}
```

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, <br> which can be configured, see Set Clear State below. The drop-down menu <br> offers the following options. <br> 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. <br> Reset Clear State <br> Resets Clear State to no selections. |
| BackReverts to the preceding logical (selection) state. Applies to value selections <br> and all the commands in the Edit menu except Copy and Copy Mode. Qlik- <br> View maintains a list of the 100 latest states. Each Back command will take |  |
| one step further back in the list. |  |

### 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. |


| Demote Sheet | Moves the active sheet one step further to the right． | ） |
| :---: | :---: | :---: |
| Sheet Properties | Opens the Sheet Properties dialog，from which it is possible to modify the active sheet． | 4 |
| Create List Box | Creates a list box to display the field of choice from the database table． | 目 |
| Create Statistics Box | Creates a statistics box，which calculates statistical entities based on the possible values of a field． | 2 |
| Create Table Box | Creates a table box，suitable for showing record－oriented information． | 囲 |
| Create Multi Box | Creates a multi box，suitable for showing different attributes． | E |
| Create Chart | Creates a chart that can be made to display fields and calculated dimen－ sions． | ，11） |
| Create Input Box | Creates an input box，suitable for displaying and entering data into Qlik－ View variables． | ＝ |
| Create Current Selec－ tions Box | Creates a current selections box，suitable for displaying the current selec－ tions directly in the layout． | v |
| Create Button | Creates a button object that performs actions in QlikView，e．g．shortcut， export etc． | － |
| Create Text Object | Creates an object for displaying text information or images． | A |
| Create Line／Arrow Object | Creates a line／arrow object，suitable for drawing a line or an arrow in the layout． | E |
| Create Slider／Calendar Object | Creates a new slider／calendar object． | （6） |
| Create Bookmark <br> Object | Creates a new bookmark object． | （8） |
| Create Search Object | Creates a new search object． | （a） |
| Create Container | Creates a new container． | （1） |
| Create Custom Object | Creates a new custom object． | （3） |
| Create Time Chart | The time chart wizard 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． | 维 |

## Format Painter

## Align Left <br> Center Horizontally

## Align Right

## Align Bottom

## Center Vertically

Align Top

Space Horizontally

Space Vertically

## Adjust Left

Adjust Top

## Document Properties

User Preferences

## Edit Module

Table Viewer

WebView Mode

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 doubleclick 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.

Aligns the active sheet objects along their left border.

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

Aligns the active sheet objects along their right border.

Aligns the active sheet objects along their bottom border.

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

Aligns the active sheet objects along their top border.

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

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

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

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

Opens the Document Properties dialog, from which it is possible to modify the settings of the current document.

Opens the User Preferences dialog, where it is possible to modify settings concerning the way the user works.

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

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

Toggles WebView mode, which uses the internal web browser in QlikView to display the document layout as an AJAX page.

### 11.5 Sheet Toolbar

```
Sheets Internal
```

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 Bookmark Toolbar

## ! Bookmarks 2010-07-05-1 - 络

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

| Add Bookmark | Saves the current selection as a bookmark. |
| :--- | :--- |
| Remove Book- | Removes the bookmark selected in the drop-down menu. |
| mark |  |

### 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 74).)
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.

## New

Rename Renames the highlighted toolbar．This command is not available for the five default toolbars．

## Delete

## Reset Resets the configuration of the highlighted toolbar to default settings．

Deletes the highlighted toolbar．This command is not available for the five default toolbars．

## Commands

| Customize |  |  |  |
| :---: | :---: | :---: | :---: |
| Toolbars Commands | tions |  |  |
| To add a command to a toolbar：select a category and drag the command out of this dialog box to a toolbar． |  |  |  |
| Categories： | Commands： |  |  |
| File | $\square$ | New |  |
| View | $\square$ | Open．．． | 三 |
| Layout | 运 | Open in Server．．． |  |
| Settings | 每 | Refresh Document |  |
| Reports |  | Open URL．．． |  |
| Object |  | Open FTP．．． |  |
| Window |  | Close |  |
| All Commands | $\square$ | Save | $\checkmark$ |
| Close |  |  |  |

## 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．

## Always show full menus

Disable this option to use shorter menus with only the most common commands.
Show full menus after a short delay
When short menus are used, enable this setting to have the full menu appear after hovering over the menu.

Reset menu and toolbar usage data

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

The following display options have been grouped under Other:

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

Windows 7. On older systems the file is found in $C: \backslash$ Documents and Settingslusername $\backslash$ Application
Data \QlikTech 1 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.
[^1]
### 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

## Show System Fields <br> Show System Variables <br> Use Sounds <br> Remove Unused Bitmaps

## Scramble Connect User Credentials

Remember Login Credentials until QlikView exits

If selected, the System Fields (page 365) will be included by default in all listings of field names.

If selected, the System Variables will be included by default in all listings of variables.

Enables the built in sound effects of QlikView.
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.

Check this alternative if the connect statement wizard should mask the connect statements by scrambling user ID and password.

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.

Shift Inhibits Macros

## Keep Progress Open after Reload

## Horizontal Scroll bar

## Vertical Scroll Bar

Resize Window to Documents

## Keep Orphaned QVD Buffers

## Flush script log after

 each writeRe-open Script Dialog After Script Execution from Dialog

## Show Getting Started

 Wizard When Creating New Document
## Purge Orphaned QVD Buffers Now

Most Recently Used Files

This setting allows the user to prevent the execution of any OnOpen macro by keeping the SHIFT key depressed while opening a QlikView document.

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

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.

As the above setting, but pertaining to the vertical.
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.

This setting overrules the normal procedure whereby any automatically created QVD Files (page 393) are purged immediately after script execution, provided that the document that created them is no longer available. This setting should normally be left unchecked.

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.

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

Opens the Getting Started Wizard which guides through the steps of creating a new document based on a single Excel sheet.

Click this button to perform a manual purge of orphaned QVD buffers that have been retained by the setting Keep Orphaned QVD buffers above.

This group is used for controlling the list of recently opened QlikView files in the File menu and on the Start Page.
Under In Menu the number of recently used files listed in the File menu can be changed. The default is 8 .
Under On Startpage the number of recently used files listed on the Start Page can be changed. The default is 64 .
These changes take effect after restart.
If the alternative Show Full Path in Menu is checked, the File menu will be widened as necessary to show full paths in the list of recently used files.
Clicking the Remove URLs button will remove any URLs from the list of most recently used files.

In this group it is possible to set the character set for the log file. The default setting is ANSI, but it is possible to check the option Unicode 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 408)). 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 108).

## 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 Scheme (page 107) (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 408)).

## 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.

Bookmark Popup Time- Check this alternative and enter a time delay to have bookmark pop-up winout (s)

Use WebView in Layout
dows automatically close after a set number of seconds.

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. <br> Include Excluded Values in Search <br> 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. <br> Preferred Search Mode <br> 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. <br> Use Previous <br> The search mode of the last completed search will be used. <br> Use Wildcard Search <br> The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search. <br> Use Fuzzy Search <br> The initial search string will be a tilde ( $\sim$ ) to denote a fuzzy search. <br> Use Normal Search <br> 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. <br> Using too high settings will degrade the performance of other processes on the computer, this may however be desirable if the computer is dedicated for QlikView. <br> 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. <br> The settings are: <br> Low <br> sets the minimum amount of memory, in percentage, to be allocated to the application/process. <br> High <br> sets the maximum amount of memory, in percentage, to be allocated to the application/process. <br> Cache <br> 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 408) 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 173) (Script), the Edit Expression Dialog (page 737) dialog (Expression in Chart) and the macro editor in the Edit Module (page 875) 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

| Normal Text | Text that does not belong to any of the categories described below. |
| :---: | :---: |
| Keywords | The keywords used in the script (described under Script Statements and Keywords (page 219)), e.g. load, select, directory, semantic, etc. |
| Comments | Comments typed in the script or chart. |
| Error | The errors that QlikView detects in the script or chart expression. |
| Highlighted Text | 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. |
| Field Names | The names of the fields to be loaded and used in charts. |
| Literals | Text to be loaded literally, i.e. as a text string (usually enclosed by single quotation marks). For the difference between field names and literals see Quotation Marks (page 281). |
| Functions | The functions used in the script (described under Aggregation Functions (page 286)), e.g. div, left, if, num, etc. and charts. |
| File Names | The name of the file from which the fields are to be retrieved. |
| Script Macro | The variables used in the script. For more information on variables, see Script Variables (page 273). |
| Aggregation Functions | The aggregations functions used in the expression (described under Aggregation Functions (page 286)), e.g. sum, min, max, etc. |
| Operators | The operators used in the script (described under Operators (page 767)), e.g. + , like, etc. and in expressions. |
| Table Labels | The labels assigned to specific tables. For more information see Table Names (page 272). |
| Set Analysis | The set analysis identifiers, modifiers and operators used in the expression (described under Set Analysis (page 799)). |

## Module

## Normal Text

## Keywords

Highlighted Text

Text that does not belong to any of the categories described below.
Visual Basic Script reserved keywords.
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

| Normal Text | Text that does not belong to any of the categories described below. |
| :--- | :--- |
| Comments | Comments typed in the chart expression. |
| Error | The errors that QlikView detects in the expression, e.g. a missing parenthesis. |
| Highlighted Text | When approaching a parenthesis with the cursor in the expression, the paren- <br> thesis, as well as the commas enclosed by it, will be highlighted. Missing <br> parentheses or commas will thus easily be detected. |
| Field Names | The names of the fields used. |
| Functions | QlikView standard function names for chart expressions. |
| File Name | The name of the file from which the fields are to be retrieved. |
| Aggregation Functions | The aggregations functions used in the expression (described under "Aggre- <br> gation Functions" on page 310 and forward), e.g. sum, min, max, etc. |
| Operators | The operators used in the expression (described under"Operators" on page 307 <br> and forward), e. g. + , like, etc. |
| Set Analysis | The set analysis identifiers, modifiers and operators used in the expression <br> (described under "Set Analysis" on page 365 and forward). |

## Base Font for All Types

The font and the font size selected in this group will be applied to all the text types. Properties such as 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:

## Function Popup Help

## Remember Cursor Position

## Copy as RTF

Check this alternative to have a small pop-up window appear when typing in a function.

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.

Check this alternative to have text copied from the editor windows stored in the Clipboard 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.

$$
\begin{aligned}
& \text { Edit } \begin{array}{l}
\text { Select a command from the list and click the Change button to customize the } \\
\text { keyboard shortcut for the command. }
\end{array} \text {. }
\end{aligned}
$$

There are two optional Key Bindings:

## Visual Basic

Visual C++

Always Use Logfiles for New Documents

## File Wizard Sample

 SizeDefault Scripting Engine

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

Sets the keyboard shortcuts to defaults familiar to the Visual $\mathrm{C}++$ debugging environment.

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.

Specifies the number of records read into the table file wizard in order to analyze a table being read.

Choose between VBScript and JScript.

## Design Grid Settings

Default Margin Unit

Default Styling Mode

## Default Sheet Object

 StyleDefault Theme

## Snap Step (mm)

Sets the distance between the snap points when the design grid is displayed.
Line Distance (mm)
Sets the distance between the grid lines when the design grid is displayed.
Here it is possible to select whether the default unit for margins in the Print: Layout page should be cm or inch.

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.

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.

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.
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 specified, new documents will be created without default theme.

## Always Show Design Menu Items

If this checkbox is marked all design menu options will be available on the context menu at all times.
If unchecked some design menu options will only be available when the Design Grid command of the View Menu (page 58) is turned on.

### 12.5 User Preferences: Objects



## User Preferences, Objects

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

Enable all con-
firmation messages

## Table Defaults

## Default Label for Others

Default Label for Total

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.

With this setting enabled, QlikView will try to preserve the 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.

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

## Calculation Progress Information

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.

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 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 361). If the result of the expression is not a valid color representation, the font color will default to black.
Additional settings are:

$$
\begin{array}{ll}
\text { Drop Shadow } & \text { If this option is checked a drop shadow will be added to the text. } \\
\text { Underline } & \text { If this option is checked the text will be underlined. }
\end{array}
$$

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 <br> 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- <br> 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) <br> 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 <br> as an image. |
| Possible Values for | Check this option to have list boxes copied as possible values rather than as <br> an image. |
| List Boxes | Check this option to have buttons copied as text rather than as an image. |
| Text for Buttons | Check this option to have text objects copied as text rather than as an image. |
| Text for Text Objects | Check this option to have current selections boxes copied as text selection <br> Selections for Current <br> Selections Box |

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 \begin{tabular}{l}
The number formatting of numeric data in QlikView may not always be com- <br>
patible with other programs due to user defined settings etc. The drop-down <br>
offers three choices in Number Formatting of numeric data that is to be <br>
exported.

$\quad$

Full Formatting <br>
Exports numeric data with its full number format, just as it is shown in the <br>
sheet objects of the document. <br>
No Thousand Separator <br>
Removes any thousand separator from numeric data. This option is rec- <br>
ommended if data is to be exported to MS Excel. <br>
No Formatting <br>
Removes all number formatting from the data and exports the raw numbers. <br>
The decimal separator will be as defined in the system settings (Control <br>
Panel). <br>
It is possible to set the default character set for export in new documents. <br>
Choose one of these option: ANSI, Unicode or UTF-8.
\end{tabular}

In the Send to Excel options group it is possible to set defaults for formatting used by the Send to Excel menu command.
$\left.\begin{array}{ll}\text { Use Regional Settings } & \begin{array}{l}\text { With this option the regional settings of the operating system are used for the } \\ \text { for Send to Excel }\end{array} \\ \begin{array}{l}\text { decimal separator when the Send to Excel command is used for exporting data. }\end{array} \\ \text { When the alternative is checked, Microsoft Excel may not be able to interpret } \\ \text { numeal data correctly. By deselecting this option, a decimal point will be } \\ \text { used, regardless of the regional settings. }\end{array}\right\}$

### 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 $\mathrm{b} / \mathrm{w}$ 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 $\mathrm{b} / \mathrm{w}$, 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 Col- If this option is checked, header columns (in pivot tables) will appear on every umns 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 email 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.

| Address | The address (URL or IP number) used for the SMTP server. |
| :--- | :--- |
| Port | The port used by the SMTP server. |
| Authentication method | Select an authentication method from the list, if authentication is required by <br> the SMTP server. |

User ID The user ID used for authentication.
Password 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

Location
Reset

The following resource locations can be modified:

## QVD Buffers

The default storage location of QVD Files (page 393) 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 583).

## 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. Read more about Section Access Management in the QlikView Server Reference Manual.

The path to the respective folder location.
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)

## Script (Allow database write and execute statements)

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

File (Save doc and export data to file)

## Confirm Launch from Macro

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

Mark this box to disable QlikView checks for scripts containing the execute command and mode is write qualifier in select statements.

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

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

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 Production IDs

This option is only available with an OEM Partner License.

| Use OEM Product IDs <br> for new documents | Enable this option if the question to use OEM Product ID should be asked for <br> 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 100) and Print: Header/Footer (page 101) 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:
Scale to _\% Check this option and enter a percentage number to increase or decrease the scale of the printout.

Fit to 1x1 Pages

Fit to _ $x_{-}$Pages

Check this option to have the printout scaled to the paper size. You may get a better result if you change the Orientation.

## 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:

| Save Settings | Click this button to save your settings for header and footer while you con- <br> tinue working with this dialog. |
| :--- | :--- |
| Print Preview | This button opens a window showing a detailed Print Preview (page 104) of <br> the current, printable object. |
| Print | 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 99) and Print: Header/Footer (page 101) property pages.

| Print Selection Stamp | In the Print Selection Stamp group the option to include the Current Selec- <br> tions in the printout is possible. The text "Selection status" will be displayed <br> in the printout, followed by a list of fields and field values. The following <br> options determine on what pages the Current Selections will be included: On <br> First Page, On All Pages - Top of Pages, On All Pages - Bottom of Pages <br> and On Last Page. |
| :--- | :--- |
| Heading Text | In the edit box specify a text to be printed before the printed sheet object. This <br> text may be a Calculated Formula (page 851). By clicking on the ... button <br> 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 $\mathrm{mm}, \mathrm{cm}$ or ". The default unit is set in the User Preferences: <br> Design page. <br> Top <br> Specifies the distance between the top of the paper and the upper border of the printed object. <br> Header <br> 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. <br> Left <br> Specifies the distance between the left edge of the paper and the left border of the printed object. <br> Right <br> Specifies the distance between the right edge of the paper and the right border of the printed object. <br> Bottom <br> Specifies the distance between the bottom of the paper and the lower border of the printed object. <br> Footer <br> 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 <br> with this dialog. |
| :--- | :--- |
| Print Preview | This button opens a window showing a detailed Print Preview (page 104) of <br> the current, printable object. |
| Print | Click this button to effect the print command and close this dialog. |

### 13.3 Print: Header/Footer



[^2]In this property page you may specify settings for Header and Footer. Additional printout settings can be made in the Print: General (page 99) and Print: Layout (page 100) 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:

| Page | Click this button or enter the code $\&[$ Page $]$ to insert the page number. |
| :---: | :---: |
| Pages | Click this button or enter the code $\&[$ Pages] 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. |
| Date | Click this button or enter the code $\&[$ Date $]$ to insert the current date. Date format can be set in the Date \& Time (page 103) dialog. |
| Time | Click this button or enter the code $\&[$ Time $]$ to insert the current time. Time format can be set in the Date $\boldsymbol{\&}$ Time dialog. |
| File | Click this button or enter the code \& [File] to insert the filename. |
| Sheet | Click this button or enter the code $\&[$ Sheet $]$ to insert the name of the sheet. This option is not available when printing reports. |
| Title | Click this button or enter the code $\&[$ Title] to insert the caption title of the printed object. This option is not available when printing reports. |
| Picture | Click this button to import a picture from the Select Image dialog. The picture will be printed as graphics in the header or footer pane. You may also manually enter the code \& [Picture=filename] where filename is the complete filename and path of the file containing the graphics. |
| Report | Pressing this button or entering the code $\&[$ Report $]$ causes the title of the report to be printed. This option is only available when printing reports. |
|  | ups allow the above settings to three panes: Left section, Center section and in the desired pane to position the cursor and then click a button or type in the |

The following buttons are also available:

| Font | This button opens the Font (page 475) dialog. |
| :--- | :--- |
| Date \& Time | This button opens the Date \& Time (page 103) dialog. |
| Default | Click this button to reset the default settings for header and footer. |
| Save Settings | Click this button to save your settings for header and footer while you con- <br> tinue working with this dialog. |
| Print Preview | This button opens a window showing a detailed Print Preview (page 104) of <br> the current, printable object. |
| Print | Click this button to effect the print command and close this dialog. |

### 13.4 Date \& Time

| Date \& Time |  | $x$ |
| :---: | :---: | :---: |
| $\square$ Use System Default | Date |  |
| Current Seltings: | Date Separator |  |
| 15:31 |  |  |
| Time |  |  |
| - 24h | Time SeparatorShow Seconds | : |
| -12h |  |  |
|  | Cancel | Help |

The Date \& Time dialog
Here the preferred display of date and time can be set.
Use System Default Enable this option to apply the system (e.g. Windows) time and date formats.
Current Settings A view of the current date and time format.

Date
Date Separator
Time In the Time group settings for the time format can be set. 24h
Enable this option to show time in the 24 hour notation.
12h
Enable this option to show time in the 12 hour notation.

## Time Separator

Choose the character to be used as time separator.
Show Seconds
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:

This Sheet
All Sheets
Draw Background

By selecting this option only the current sheet will be printed.
By selecting this option all sheets in the document will be printed.
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.

| Print | Transfers control to the Print: General (page 99) dialog, from where you can print the <br> active object. |
| :--- | :--- |
| \#pages drop- <br> down | This drop-down menu offers a quick way to change which pages to display in the pre- <br> view. |
| Previous | If the preview contains several pages, you can click this button to view the previously <br> shown page. |
| Next | If the preview contains several pages, you can click this button to view the next page. |
| Add Page | Extends the preview to include another page if the active object does not fit in one <br> page. |
| Remove Page | Removes the current preview page. |
| Close | Closes this dialog. |
| Help | 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 104) 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 104) 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.

## Clipboard Format In this group, the format of the copy list is set. Column, Comma Delimited

Line and Tab Delimited Line are the available options.

| Quoting | In this group, the quoting for the selected elements can be set. <br> Quote with ' will enclose all elements with single quotes. This is useful if the <br> elements copied are to be pasted into the script as field values. <br> Quote with " will enclose all elements with double quotes. This is useful if <br> the elements copied are to be pasted into the script as field names or into a Vis- <br> ual Basic script, e.g. a QlikView macro. <br> None will leave the elements unquoted. <br> OK <br> Closes the Clipboard Copy List dialog and transfers its content to the Win- <br> dows Clipboard. |
| :--- | :--- |
| Closes the Clipboard Copy List dialog without transferring its content to the |  |
|  | Windows Clipboard. |

### 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

### 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, i.e. selected. The selection may affect the states of a large number of values in the other sheet objects. White cells represent optional field values, whereas gray cells represent field values excluded by selections. When a selection is made, natural inner joins are dynamically created between all the associated tables.
Selections can be made not only in list boxes, but also in statistics boxes, multi boxes, table boxes and charts. See below for more information about selections in objects.
To cancel a previously made selection, click it or choose Clear from the Object menu, or choose one of the Clear commands in the Selections menu. When an excluded value is clicked, selections in conflict with this value are canceled, and the value clicked becomes selected.

### 14.2 Color Scheme

The selection state of a field value is shown by the color in the cell.
The table below lists the colors of the default (Classic) color scheme. These colors can be modified if one of the other color schemes is chosen in the Selection Appearance group on Document Properties: General page.
The state of a field value is shown by the color of its cell. The following color scheme is used:

| State | Background |
| :--- | :--- |
| Select | Green |
| Optional | White |
| Alternative | White/Yellow |
| Locked | Blue |
| Excluded | Gray |
| Forced excluded | Red |

When the Show Alternatives (page 462) option is enabled, QlikView displays non-selected cells in the corresponding list box as alternative values (white), unless they are excluded by selections in other list boxes. They are however logically excluded, and not included in calculations made on possible (optional and selected) values. When the Show Alternatives option is 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 data and making selections in list boxes and multi boxes. The QlikView Classic, Corner Tag and LED styles all use color coding for indication of selected, possible and excluded values. The Windows Checkboxes style mimics the standard Windows interface with 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 is used can be controlled via settings on the Document Properties: General page (Document Properties: General (page 408)), the User Preferences: General page (User Preferences (page 77)) and the list box properties: Presentation page (List Box Properties: Presentation (page 471)). Representation of Logical State
The logical state of data in QlikView is in the color-coded QlikView Classic style represented by color coding of background and text in the cells of list boxes and multi boxes. In some other sheet objects and on the sheet tabs so called selection beacons may appear to indicate selections. The \& and ! characters may appear to the left of data in list boxes and multi boxes in order to indicate AND and NOT selections.
With the Windows 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 color- | Windows check |
| :--- | :--- |
| coded styles | box styles (icons) |

QlikView State

| Possible | White/Black | $\square$ |
| :--- | :--- | :--- |
| Excluded | Gray/Light Gray |  |
| Selected | Green/Black | Gray/Light Gray |
| Select Excluded | $\mathbf{\checkmark}$ |  |
| Locked | Blue/Yellow | $\mathbf{Q}$ |
| Locked Excluded | Gray/Light Gray | $\mathbf{Q}$ |
| And Selected | Green/Black <br>  | $\mathbf{8}$ |
| Not Selected | Red/Black with ! | $\mathbf{!} \times$ |

## 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 selections under the Windows and LED Check Boxes style are considered toggle selections, i.e. they work as if the ctrl key is hold down while making selections under the color-coded styles.
2. All list boxes are treated as if the Show Alternatives option is selected, regardless of the actual setting of this property.

### 14.4 Indicator

Indicators (or beacons) are colored dots that sometimes appear in tables, on tabs and in the right-hand corner of the Status Bar (page 75). 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 86) dialog.
Selection indicators will appear in the current selections box as well as in the free-floating current selections text box, in order to distinguish between selected and locked values.
The color of the indicator follows the general color scheme:
A green dot for selected values.
A blue dot for locked selections.
A red dot for de-selected values in AND-mode.

### 14.5 Multiple Selection Within a Field

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

- Position the cursor in the list box, then drag it over a number of field values while pressing the mouse button.
- Click the first field value to be selected, then CTRL-click each additional selection.
- Click the top item to be selected, then SHIFT-click the bottom item to be selected. This way all the items in between will be selected. However, if the sort mechanism is on, the first selection may cause the order to change, which makes it difficult to make a correct second selection.
- Confirm a text search (see below) by hitting the ENTER key. This will result in all matching field values being selected. By keeping the CTRL key depressed while pressing ENTER the selections from the text search will be added to previous selections.
To deselect a previously made additional selection, CTRL-click it.
If a multiple selection is made 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.
A multiple selection can be interpreted in two different ways, either as a logical or or as a logical and. Default is logical or, i.e. QlikView will find a solution that is associated to one or more of the selected field values.


### 14.6 Moving Selections

The current selection in an active list box or in an opened multi box field can be moved by means of the keyboard keys. The following keys may be used:

$\downarrow \quad$| 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. |

PgDn | 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. |

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.
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.

## Example:

Assume that the list box ( E and F selected) is as follows:

$\downarrow \mathrm{F}$ and G will be selected.
$\uparrow D$ and E will be selected.
PgDn G and H will be selected.
PgUp C and D will be selected.
Home A and B will be selected.
End I and J will be selected.

### 14.7 Locking Selected Field Values

After selecting an excluded field value, previous selections in conflict with the new choice are canceled. To prevent a previous selection from getting canceled this way, 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

| Q Current Selections |  | 5 |
| :---: | :---: | :---: |
| Fields | Status | Values |
| Model | - | Alfa Romeo 1471.61 ... |
| Engine | - | Petrol |
| Transmi... | $\bigcirc$ | Manual |

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 Indicator (page 109) are used for distinguishing between selected and locked values. It is opened from the View menu or the Current Selections button in the toolbar.
Right-clicking in the Current Selections window without having highlighted a field, the float menu contains the following commands:

Clear All Selections Deselects all the selected values of the document.
Lock All Selections Locks all the selected values of the document.
Unlock All Selections Unlocks all the locked values of the document.
Data
Copies the name(s) of all the selected value(s) of the document, as well as the name(s) of the field(s) in which the selections were made, to the clipboard.
The Current Selections window resembles a sheet object in the way that it is sizable and can be kept open while working with a document. It will be left open when quitting QlikView and reappear the next time QlikView is started. As the sheet objects, it is dynamically updated as soon as new selections are made, thus allows to always have an overview of the current selections.
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 77)). 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

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

## Statistics Boxes

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

## Multi Boxes

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

## Table Boxes

Selections can be made by clicking in any cell or by painting over an area covering one or more rows and one or more columns. If the option Drop-down Select is active, an arrow is displayed in the column heading. Clicking on the arrow displays a list of the values that belong to this field. Selections and searches can be made in this list just as in a list box.

## Slider/Calendar Objects

In sliders, where a single field is the basis, it is possible to select a value by adjusting the thumb tack to the desired position. If the slider is accordingly configured, the size of the thumb tack can be changed with a mouse click. This way it is possible to select several values.
Clicking on the small calendar symbol in a calendar object opens the calendar. There it is possible to select a date or a whole period with the mouse depending on the configuration of the calendar object, and this selection is transferred to the underlying field. Using CTRL+CLICK it is possible to select several periods even if they are in different months or years.

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

Selections can be made inside the plot area by clicking on a single data point or painting over several data points. When painting, the area covered is shown with a green raster. The selection will be made for the dimension values used to calculate the selected data point(s).
Selections can be made by clicking or painting in the chart legend (except when the legend is indicating chart expressions rather than dimension values).
Selections can be made by clicking or painting over the dimension axes and its labels (except scatter charts). The corresponding field values will be selected.
Selections can be made by painting over the expression axes and its labels. The field values which generate data points in the indicated result area will be selected. When painting selections in line charts and bar charts with more than one dimension the behavior of the QlikView selection logic differs slightly from 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

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

## Block Charts

In block charts individual blocks can be selected. With selections the drilldown functionality comes into play. Selecting the first block refers to the first dimension, selecting a second block within the first refers to the second dimension etc.
It is also possible to select several blocks by painting an area. This selected area is marked green until the mouse button is released. Such a selection refers back to the value or values of the first dimension. Based on these values the corresponding blocks are calculated. If a selection crosses the block borders of several values that belong to the first dimension, it will affect all related values belonging to the second and third dimensions as well, not only those in the selected area.

## Gauge charts

Selections cannot be made in gauge charts.

## Straight Tables

Selections can be made in the dimension columns by clicking on a cell or painting over several cells. The selected area is marked green until the mouse button is released.
If Drop-down Select is activated in a column that represents a dimension, a small arrow will be displayed in the column header. Clicking on the arrow will display a list of all values of the field. Selections and searches can be made in this list.
Selections can be made in the expression columns by clicking in a single cell. The selection will be made for the dimension values used to calculate the selected expression cell.

## Pivot Tables

Selections can be made in the dimension columns/rows by clicking on a single cell. The selected cell is marked green until the mouse button is released.
If Drop-down Select is activated in a column that represents a dimension, a small arrow will be displayed in the column header. Clicking on the arrow will display a list of all values of the field. Selections and searches can be made in this list.
Selections can be made in the expression columns/rows by clicking in a single cell. The selection will be made for the dimension values used to calculate the selected expression cell.

### 14.10 Search

Selections can be made through text search as well. To enter a search string, click on the list box header, then simply type the search string. The string is not case sensitive. The search string will appear in the popup search box. As a result, QlikView will display all the values of the selected field that fulfill the criteria of the search string. When pressing ENTER, or clicking on one of the cells in the result, the value(s) become(s) selected. By keeping the CTRL key depressed while pressing ENTER the selections from the text search will be added to previous selections.
The search box will close automatically when ENTER or ESC is hit or clicking in the layout. It is also possible to close it by clicking the icon in the search box. The search box is sizable and will retain its size when it is opened again.
If several list boxes are active (SHIFT-click on their headers to make them active), they are all included in the text search. Pressing ENTER to select the resulting values is however not possible until there are optional values in only one of the active list boxes. If a selection has already been made, there are two ways in which a search can be interpreted: either searching only among the optional values, or searching among all values, i.e. include 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.
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
typed. $\square$
If no wildcards are used (Normal Search), 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.
The Search string may however contain wildcard characters (Wildcard Search). 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.
$\wedge$ whole string.
The ENTER key can be pressed to select the values found, and the ESC key can be pressed to cancel the operation.


## Examples:

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

## Fuzzy Search

If starting the text search with a tilde sign $\sim$ the text search window will be opened in fuzzy search mode. The search window will contain the tilde with the cursor placed after it. When typing 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 ENTER is hit the first value in the list will be selected.

## Numeric Search

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

## Examples:

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

## Initial Search Mode

When starting to type text, the behavior may differ: in some cases, QlikView add wildcards (to facilitate a Wildcard Search) or a tilde (for Fuzzy Search) to the search string; in some cases QlikView will not add
any characters at all (for Normal Search) to the search string.
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. If the search string begins with a tilde, a Fuzzy Search will be made. If the search string contains neither wildcard characters, nor a leading tilde, a Normal Search will be made.
It is always possible to change the search mode simply by deleting or adding wild cards, a tilde ( $\sim$ ), a greater than $(>)$ or smaller than $(<)$ symbol in the search string.

## Associated Search

The search box contains a chevron to the right. If this is clicked, 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 a selection in the secondary list has been made, it is possible to enter a new search string before making the selection in the primary list. Finally, when a selection is made 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 (see below), which can be invoked by the keyboard shortcut CTRL+SHIFT+F. If starting the text search with an equal sign $=$, it is possible to enter an advanced search expression involving search criteria for associated fields and full boolean logic. After the equal sign it is possible to type any valid QlikView layout expression (Edit Expression Dialog (page 737)). The expression will be evaluated for each field value in the search field. All values for which the search expression returns a non-zero value will be selected.

## Examples:

$=$ MyField like 'A*' or MyField like ${ }^{\prime *}$ Z'
if e.g. invoked from a list box containing the field MyField, the search would return all field values either starting with the letter A or ending with the letter Z .

## $=$ sum(Sales) $>$ sum(Budget)

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

### 14.11 Advanced Search Dialog



## The Advanced Search dialog

This dialog provides a convenient way of formulating complex search queries to be applied on QlikView fields. Unlike the standard Search (page 113) 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.
\(\left.$$
\begin{array}{ll}\text { Search in } & \begin{array}{l}\text { The field in which the search will be made. When entering the dialog it } \\
\text { will be set to the field of the active list box. It is possible to change search } \\
\text { fields by using the drop-down at any time. }\end{array} \\
\text { Search Expression } & \begin{array}{l}\text { This is where the search expression is typed. The same rules apply as when } \\
\text { using the normal search box. }\end{array} \\
\text { Go } & \begin{array}{l}\text { Applies the search to the search field }\end{array} \\
\text { QlikView remembers the last } 100 \text { selections. Click this button to go one } \\
\text { step back in the list of selections. }\end{array}
$$ \quad \begin{array}{l}Click this button to go one step forward in the list of selections (equivalent <br>
to canceling the last Back command). This only possible if the Back com- <br>

mand has been used immediately before.\end{array}\right]\)| Clears selections in the current search field. |
| :--- | :--- |

In the bottom pane of the dialog three tabs are found, which can help when building advanced search expressions.

## Fields

On the Fields page the controls for pasting syntax relating to QlikView field data are found.

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

## Functions

On the Functions page the controls for pasting syntax relating to QlikView general functions are found.

| Function Category | In the drop-down it is possible to select a category of functions in order to <br> make navigation in the Function Name drop-down easier. |
| :--- | :--- |
| Function name | In the drop-down it is possible to select a function for pasting into the expres- <br> sion out of all functions available in the QlikView layout. The list can be <br> reduced to show only functions belonging to a certain category by means of a <br> 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 page there is a pane showing the argument syntax of the function selected in the Function Name drop-down.

## Variables

On the Variables page the controls for pasting syntax relating to QlikView variables are found.
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 page there is a pane showing the current value of any variable selected in the Variables drop-down.

### 14.12 And-Mode in List Boxes

A multiple selection within a field is by default interpreted as logical OR, which means that data associated to any of the selected field values will be shown in other fields. It is however possible to set certain list boxes to And mode instead. When multiple selections are made in a list box set to And mode, the associations in other fields must be to all selected values, which is exactly how associations are normally made to multiple selections in several fields.
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-Tables

A field cannot always be set to logical And mode. The reason for this is that the and alternative is logically meaningful only if the concerned field is linked to only one other field.
The And mode option is only allowed under very strict conditions, imposed by the theory behind the QlikView logic. In order to be used in And mode, the following criteria must be fulfilled:

- the field must only exist in one internal table,
- the field must be the second column of no more than two columns, and
- the table must not contain any duplicate records, and
- the table must be loaded using a distinct qualifier. If the table is loaded using a Select statement, a preceding Load distinct * must be used.


## Forced Exclusion/Not Selection

Closely related to the and selection is the forced exclusion, or not selection. Here it is possible to explicitly exclude a field value, i.e. the solutions found by QlikView may not be associated with the excluded value.
The not selection is made by clicking a cell and keeping the mouse button depressed until the cell turns red. CTRL-clicking in this manner is equivalent to an exclusion that is an additional requirement to the previous selections/ exclusions. The not selection can only be made on a field that is in And mode.

### 14.13 Loosely Coupled Tables

A loosely coupled table is one where the normal QlikView logic has been disconnected internally. This means that selections in one field do not propagate through to the other fields in the table. This chapter will show some examples of how loosely coupled tables alter the QlikView logic.

## A Basic Example

Have a look at the following three table boxes, each one representing a table read into QlikView:

| Tab1 |  |
| :--- | :--- |
| $B$ | $A$ |
| 1 | $x$ |
| 2 | $y$ |
| 3 | $z$ |


| Tab2 |  |
| :--- | :--- |
| $A$ | $C$ |
| $x$ | 6 |
| $y$ | 7 |
| $z$ | 8 |


| Tab3 |  |
| :--- | :--- |
| C | D |
| 6 | a |
| 7 | b |
| 8 | C |

If the value 2 is selected in field $B$ the following will happen:


The selection ripples through all tables. Now let us keep this selection but make Tab2 loosely coupled. This means that the logic will be cut between the fields A and C in Tab2. The result will look this:


Note that Tab2 shown here is a table box and not the table itself. The table box will show all possible combinations between the fields of its columns. Since there is no logic between fields A and C all combinations of their respective possible values are shown.

## Avoid Circular Reference

The next example shows how loosely coupled tables can be useful for avoiding circular references in the data structure:


This data structure is actually not a very good one since the field name Country is used for two different purposes. In one table it indicates where the car owner lives and in another it shows where the car manufacturer resides. With the data in the tables an impossible logical situation is faced. Wherever a selection is made, it is possible to follow associations leading through every single cell in all three tables.
A decision whether the home country or the car manufacturing country is most important has to be made. If the Carmaker table is loosely coupled, the associations from Cadillac to USA and from Volvo to Sweden will be broken. By clicking Sweden it will associate with Björn Borg and Cadillac. By clicking Volvo it will associate with George Bush and USA.
If it is prefered to concentrate on car manufacturers it would make sense to make the Home table loosely coupled instead.

## Another Example

Let us have a look at another common situation where loosely coupled tables can be useful. Below three tables are found in a rather typical structure: one transaction table and two dimension tables associating to it via one field each.

| Time |  |
| :--- | :--- |
| Year | Date |
| 1999 | $1999-11-30$ |
| 1999 | $1999-12-01$ |
| 2000 | $2000-01-15$ |
| 2000 | $2000-01-22$ |


| Trans |  |  |
| :--- | :--- | :--- |
| Date | Prod | Amount |
| $1999-11-30$ | A | 10 |
| $1999-11-30$ | E | 14 |
| $1999-12-01$ | A | 15 |
| $1999-12-01$ | B | 11 |
| $2000-01-15$ | C | 17 |
| $2000-01-15$ | D | 13 |
| $2000-01-22$ | B | 16 |
| $2000-01-22$ | C | 12 |


| Product |  |
| :--- | :--- |
| Proc | ProdGrp |
| A | X |
| B | X |
| C | X |
| D | Y |
| E | $Z$ |

Now, let us say a pivot table showing sales per year and product group is wanted. If we just create one beside two list boxes showing the dimension fields it will look like this:

| Sum(Amount) |  |  |
| :--- | :--- | ---: |
| Year | ProdGrp | Sum(Amount) |
| 1999 | $X$ | 36 |
|  | $Z$ | 14 |
|  |  | 50 |
| 2000 | $X$ | 45 |
|  | $Y$ | 13 |
|  |  | 58 |
|  |  | 108 |


| Year | ProdGrp |
| :--- | :--- |
| 1999 | $X$ |
| 2000 | $Y$ |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Even though this is a correct pivot table, the effects of QlikView's logic could here potentially lead to undesired results. If we select the year 2000 the layout will look as follows:

| Sum(Amount) |  | $-\square$ |
| :--- | :--- | ---: |
| Year 0 | ProdGrp | Sum(Amount) |
| 2000 | $X$ | 45 |
|  | $Y$ | 13 |
|  |  | 58 |
|  |  | 58 |



Product group Z is "gone". This is natural since the value Z in the field ProdGrp has been excluded by the selection of the value 2000 in the field Year. However, the boss is likely to want Z in the chart with a 0 in the sum(Amount) column, so that it is clear to everyone that product group Z exists and that nothing has been sold in 2000.

One could argue that the two fields Year and ProdGrp really have nothing to do with each other and therefore should not interact just because they happen to be associated via the Trans table. So let's fix that by declaring the Trans table as loosely coupled. Immediately our little layout will have the following look of:


The table looks like we want it to look. Note that the selection in the Year list box does not render any value in the ProdGrp list box excluded.
As a summary can be said that this situation, with one or more transaction tables surrounded by a number of dimensions which should not be excluded, is quite common. Loosely coupled tables is one possible way of handling such cases.

## Loosely Coupled Tables and Subtotals

When using loosely coupled tables in combination with non-hierarchical dimension data subtotals in pivot tables may become incorrect. The example below is very similar to the previous example, but product B now belongs to two product groups, $X$ and $Y$. The product $D$ has disappeared and that sale is now product $B$ instead.

| Time |  |
| :--- | :--- |
| Year | Date |
| 1999 | $1999-11-30$ |
| 1999 | $1999-12-01$ |
| 2000 | $2000-01-15$ |
| 2000 | $2000-01-22$ |


| Trans |  |  |
| :--- | :--- | :--- |
| Date | Prod | Amount |
| $1999-11-30$ | A | 10 |
| $1999-11-30$ | E | 14 |
| $1999-12-01$ | A | 15 |
| $1999-12-01$ | B | 11 |
| $2000-01-15$ | B | 13 |
| $2000-01-15$ | C | 17 |
| $2000-01-22$ | B | 16 |
| $2000-01-22$ | C | 12 |


| Product |  |
| :--- | :--- |
| Prod | ProdGrp |
| A | X |
| B | X |
| B | Y |
| C | X |
| E | $Z$ |

This means that the total amount sold is the same, which can be seen if we take a look at the pivot table, now expanded with Prod as a third dimension:

| Sum(Amount) |  |  | $\square \square$ |
| :---: | :---: | :---: | :---: |
| Year | ProdGrp | Prod | Sum(Amount) |
| 1999 | X | A | 25 |
|  |  | B | 11 |
|  | Y | B | 11 |
|  | Z | E | 14 |
|  |  |  | 50 |
| 2000 | X | B | 29 |
|  |  | C | 29 |
|  | $Y$ | B | 29 |
|  |  |  | 58 |
|  |  |  | 108 |

As you can see from the absence of product group Z in the year 2000 we have not set any table loosely coupled. QlikView handles all subtotals correctly, i.e. the two occurrences of $B$ under each year are only counted once in the subtotals. This is an effect of QlikView's normal internal logic.
Now let us loosen the Trans table as we did before. The pivot table will look as follows:

| Sum(Amount) |  |  | $\square \square$ |
| :---: | :---: | :---: | :---: |
| Year | ProdGrp | Prod | Sum(Amount) |
| 1999 | $\times$ | A | 25 |
|  |  | B | 11 |
|  |  | C | 0 |
|  | Y | B | 11 |
|  | Z | E | 14 |
|  |  |  | 61 |
| 2000 | X | A | 0 |
|  |  | B | 29 |
|  |  | C | 29 |
|  | Y | B | 29 |
|  | Z | E | 0 |
|  | Z |  | 87 |
|  |  |  | 148 |

By cutting the association within the Trans table there is now no way for QlikView to keep track of what has been counted before in the subtotals. The two occurrences of $B$ will be counted twice and the subtotals are too big.

This situation should be avoided, which leads to the conclusion that loosely coupled in combination with non-hierarchical dimensions should be used with care.

## Note!

The Total Mode (as set on the chart's Expressions page) is set to Expression Total (default) in all the cases above. If the Sum of Rows mode is used there is no difference between the two cases.

### 14.14 Alternate States

## Description

QlikView allows users to detach a chart from the underlying data model. When a chart is detached, the object no longer responds to user selections. When the user re-attaches the object, it instantly updates to reflect the current state of the selections.
Alternate States are similar to this. 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 are not accessible in the load script. They are a UI feature.
The developer invokes the Alternate States dialog from Alternate States... button on the Document Properties: General tab. The QlikView developer can create any number of states within the QlikView document and provides a name for each state created. Once Alternate States functionality is enabled, the QlikView developer can also create new 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.

## Note!

Alternates States functionality is enabled by a QlikView developer and should be used with caution as it can cause great confusion with end-users.

## State Identifiers

The state names created by the QlikView developer are known as State Identifiers. 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.

## Logical Behavior when Using Alternate States

Pressing the Clear button affects all states. However, a menu item has been added to the Clear button dropdown menu that allows a user clear the selections of 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.

## Features in Alternate States

The following features are included in the Alternate States:

- 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.
- 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.
- Variables belong to the default state. Changes in other states will not affect variable values.
- Triggers fire in all states.
- 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.
- 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>.


## 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

## User bookmarks

## Personal Server bookmarks

## Shared Server bookmarks

are stored inside the qvw document. They will always be available to whoever opens the document locally or from a QlikView Server.
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.
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.
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.

| Add Bookmark |  |
| :--- | :--- |
| Bookmark Name: |  |
| My Bookmark |  |
| $\nabla$ Make this bookmark a document bookmark |  |
| $\square$ Share Bookmark with Other Users |  |
| $\nabla$ Include Selections in Bookmark |  |
| $\square$ Make bookmark apply on top of current selection |  |
| $\square$ Include Layout State |  |
| $\square$ Include Scroll positions |  |
| $\square$ Include Input Field Values |  |
| Info Text: |  |
| $\square$ Pop-up Message |  |
| $\square$ | OK |

The Add Bookmarks dialog for local documents
$\left.\begin{array}{ll}\text { Bookmark Name } & \begin{array}{l}\text { The default name for the created bookmark is that of the current date. In addi- } \\ \text { tion, the first bookmark created on a specific day gets number 1, the second } \\ \text { number 2, etc. However, it is possible to change the default name to a more } \\ \text { explanatory text by entering an optional name. }\end{array} \\ \text { Make this bookmark a } \\ \text { document bookmark } & \begin{array}{l}\text { By marking the check box, the bookmark will be saved with the document. } \\ \text { This option is available for local documents only. If the check box is not } \\ \text { marked, the bookmark will be created as a user bookmark, stored locally on } \\ \text { the computer. }\end{array} \\ \text { Include Selections in } & \begin{array}{l}\text { By marking the check box the bookmark will include the selections made in } \\ \text { the application. }\end{array} \\ \text { Bookmark } \\ \text { Make bookmark apply } & \begin{array}{l}\text { By marking the check box, the bookmark will be applied without first clearing } \\ \text { existing selections in the document. }\end{array} \\ \text { on top of current selec- }\end{array} \quad \begin{array}{l}\text { By marking the check box, the bookmark will store information about the }\end{array}\right\}$

## The differences for server documents are as follows:

| Add Bookmark |
| :--- | :--- |
| Bookmark Name: |
| My Server Bookmark |
| $\nabla$ Create as a Server Bookmark |
| $\square$ Share Bookmark with Other Users |
| $\nabla$ Include Selections in Bookmark |
| $\square$ Make bookmark apply on top of current selection |
| $\square$ Include Layout State |
| $\square$ Include Scroll positions |
| $\square$ Include Input Field Values |
| Info Text: |
| $\square$ Pop-up Message |
| $\square$ |
| OK |
| Cancel |

## 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 125).
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:

| Show | Enable this option if the bookmark should appear in bookmark object listings <br> and in the Bookmarks menu. If the check box is left unchecked, the bookmark <br> will not appear in those places, but will remain accessible via this dialog. |
| :--- | :--- |
| Name | Name of the bookmarks. To sort the bookmarks in alphabetical order, click the <br> header of the column. If clicked twice, the bookmarks are sorted in reverse <br> alphabetical order. |
| + | If the check box is marked for a bookmark, that bookmark will be applied <br> without first clearing existing selections in the document. A bookmark can <br> only be applied on top of existing selections if they do not conflict with the <br> selections in the document. |


| Layout | If the bookmark was created complete with layout information, use the check- <br> box to toggle the layout setting for the bookmark. When the layout setting is <br> on, the program will attempt to recreate the layout that prevailed when the <br> bookmark was created. This includes switching to the correct sheet and updat- <br> ing the layout of any sheet objects on that sheet. |
| :--- | :--- |
| This setting cannot be used to add layout information to a bookmark that was |  |
| originally created without it. |  |
| Created | Timestamp information on when bookmarks were created. |
| IDe unique ID of the bookmark. Upon creation, every bookmark is assigned a |  |
| unique ID for control via Internal Macro Interpreter (page 873). 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. |  |$\quad$| A bookmark ID may be changed via Internal Macro Interpreter (page 873) 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.). |

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.
Select Displays the selected bookmark on the screen.

Replace

## Rename

Set Clear State

Remove Removes the selected bookmark. Only bookmarks that you have created can be removed.

Clear All Removes all bookmarks in the current pane of the dialog. Only bookmarks that you have created can be removed.
Replaces the selected bookmark with the current selections. Only bookmarks that you have created can be replaced.

Opens the Rename Bookmark dialog where it is possible to specify a new name for the bookmark. Only bookmarks that you have created can be renamed.

Click this button if you want to use the selection of the highlighted bookmark as clear state.

| Edit Info | Here it is possible to edit the info text. |
| :--- | :--- |
| Import | After browsing for and selecting a previously saved bookmark (.qbm) file, the <br> Import Bookmarks (page 129) dialog will open to import bookmarks. |
| Export | Opens the Export Bookmark(s) (page 129) dialog where it is possible to <br> export selected bookmarks to a QlikView bookmark (.qbm) file. |
| Promote | Moves the selected bookmark one step up in the list. It is also possible to do <br> this by clicking and dragging it to any position in the list. |
| Demote | Moves the selected bookmark one step down in the list. |
| Move Local User Book- |  |
| marks to Server | This text only appears when working with a server document and in the User <br> Bookmarks tab. By clicking on the text it is possible to convert all the local <br> user bookmarks to server bookmarks, stored in the server repository. This is a <br> recommended practice as the personal server bookmarks will stay accessible <br> 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 con- |  |
| version 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

## Current Bookmarks

## Import

## Rename

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

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.

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

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

| Rename Bookmark |  |
| :--- | ---: |
| New name for 2010-06-21-1 |  |
| Sales | OK |

### 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:

$\boldsymbol{\operatorname { s u m }}(\{[$ 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.

Document Reports

## User Reports

## My Server Reports

## Shared Server Reports

are created with the QlikView document and stored as part of the QVW file. Any user accessing the QlikView document locally or via QlikView Server can access the document reports in the document.
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).
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.
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 134) 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 134).

### 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:

## 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:

| Show | Enable this option if you want the report to appear in the Reports menu. If the <br> check box is left unchecked, the report will not appear but will still remain <br> accessible via this dialog. |
| :--- | :--- |
| Name | Name of the report. |
| ID | The unique report ID (see below). |
| Pages | Number of pages currently defined in the report. |
| Share | This check box is only available in the My Server Reports list. By marking it <br> for one of the My Server Reports, the report will be available to other authen- <br> ticated 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 |  |
| Ahared Server Reports list. Deselect the check box at any time to revoke the |  |
| sharing. |  |$\quad$| The authenticated name of the creator of the report. |
| :--- |

## Move Local User Reports to Server

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:

[^3]
## Edit Menu

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

## Reports Menu

Add
Delete

| Promote | Moves the selected report up one step in the list of reports. Only available <br> when viewing the Report List. |
| :--- | :--- |
| Demote | Moves the selected report down one step in the list of reports. Only available <br> when viewing the Report List. |
| Export | Saves the report file as an XML document. Only available when viewing the <br> Report List. |
| Import | Imports a report that has been saved as XML. Only available when viewing <br> the Report List. |
| Print Preview... | Opens the Print Preview dialog and where you can see how the current report <br> will look when printed with the current selections in the QlikView document. |
| Print... | Opens the Print dialog and prints the current report. |
| Report Settings | Opens the Report Editor: Report Settings Dialog (page 141) dialog where <br> you can set various properties for the report. This includes margins, head- <br> er/footer setting etc. |

## Page Menu

| Promote | Promotes the currently selected page. This may also be achieved by means of <br> dragging and dropping the page in the page list pane. |
| :--- | :--- |
| Demote | Demotes the currently selected page. This may also be achieved by means of <br> dragging and dropping the page in the page list pane. |
| Add Multi Page | Adds a multi-paper page after the currently selected page. |
| Add Single Page | Adds a single-paper page after the currently selected page. |
| Delete | Deletes the currently selected page. |
| Page Settings | Opens the Report Editor: Page Settings Dialog (page 144) dialog where you <br> can set various properties for the selected page. |

Adds a report. Only available when viewing the Report List.
Deletes a report. Only available when viewing the Report List.
Moves the selected report up one step in the list of reports. Only available when viewing the Report List.

Moves the selected report down one step in the list of reports. Only available when viewing the Report List.

Saves the report file as an XML document. Only available when viewing the Report List.

Imports a report that has been saved as XML. Only available when viewing the Report List.

Opens the Print Preview dialog and where you can see how the current report will look when printed with the current selections in the QlikView document.

Opens the Print dialog and prints the current report.
Opens the Report Editor: Report Settings Dialog (page 141) dialog where er/footer setting etc.

Promotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.

Demotes the currently selected page. This may also be achieved by means of dragging and dropping the page in the page list pane.

Adds a multi-paper page after the currently selected page.
Adds a single-paper page after the currently selected page.
Deletes the currently selected page.
Opens the Report Editor: Page Settings Dialog (page 144) dialog where you can set various properties for the selected page.

## Item Menu

\(\left.$$
\begin{array}{ll}\text { Left Align } & \begin{array}{l}\text { Only available when two or more print objects are selected (with the } \\
\text { help of CTRL-click). Aligns the selected objects to the left. Also avail- } \\
\text { able as right-click command in the page preview pane. }\end{array} \\
\text { Center Horizontally } & \begin{array}{l}\text { Only available when two or more print objects are selected. Centers the } \\
\text { selected objects horizontally. Also available as right-click command in } \\
\text { the page preview pane. }\end{array}
$$ <br>
Only available when two or more print objects are selected (with the <br>
help of CTRL-click). Aligns the selected objects to the right. Also avail- <br>

able as right-click command in the page preview pane.\end{array}\right\}\)| Only available when two or more print objects are selected (with the |
| :--- |
| help of CTRL-click). Aligns the selected objects to the bottom. Also |
| available as right-click command in the page preview pane. |


| New Text... | Creates a new text object (in a hidden location in the document) and <br> then opens the Text Object Properties dialog where the properties for the <br> new text object can be edited. The text object will appear as a regular <br> print item in the page preview pane, where it can be moved and sized <br> just like any other print item. |
| :--- | :--- |
| New Current Selections... | Creates a selection stamp object for the report. The selection stamp will <br> appear as a regular print item in the page preview pane, where it can be <br> moved and sized just like any other print item. |
| Delete | Deletes the currently selected item(s). Also available as right-click com- <br> mand in the page preview pane. |
| Item Settings... | Opens the Report Editor: Item Settings Dialog (page 147) dialog where <br> you can set various properties for the selected print item. Also available <br> 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:

| Page Settings | Opens the Report Editor: Page Settings Dialog (page 144) dialog where <br> you can set various properties for the selected print item. Also available <br> as right-click command in the page preview pane. |
| :--- | :--- |
| Delete | Deletes the highlighted page. |
| Add | Adds a single-paper page or a multi-paper page after the currently <br> selected page. |
| Item Settings | Opens the Report Editor: Item Settings Dialog (page 147) dialog where <br> you can set various properties for the selected print item. Also available <br> as right-click command in the page preview pane. |
| Delete | Deletes the highlighted item. |
| Add | 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:

| Add Single Paper Page | Adds a single paper page after the currently selected page. |
| :--- | :--- |
| Add Multi Paper Page | Adds a multi paper page after the currently selected page. |
| Delete Page | Deletes the currently selected page. |
| Promote Page | Promotes the selected page one step up the list. It is also possible to drag <br> and drop the page in the page list pane. |
| Demote Page | Demotes the selected page one step down the list. It is also possible to <br> 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.
Align Top
Center Vertically

Align Bottom

Align Left

Center Horizontally

Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the top.

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.

Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the bottom.
Only available when two or more print objects are selected (with the help of CTRL-click). Aligns the selected objects to the left.

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.


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

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

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

| Print Preview | Push this button to open the Print Preview dialog and see how the cur- <br> rent report will look when printed with current selections in the Qlik- <br> View document. This button is also available from the Page Editor page <br> of the Report Editor dialog. |
| :--- | :--- |
| Print | Push this button to open the Print dialog and print the current report. <br> This button is also available from the Page Editor page of the Report <br> Editor dialog. |
| Copy | Copies the highlighted report, page or item. |
| Paste | 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:
Name $\quad$ Name of the report. This may be any text string. The name may be given as a calculated formula for dynamic update.

The unique ID of the report. Upon creation, every QlikView layout entity, including reports, is assigned a unique ID for control via Automation. The ID by default consists of a combination of letters defining the type of entity, as well as of a number. The first report of a document will be assigned the ID RP01. An ID may be changed to any other string that is currently not used for any other report, sheet object, sheet, bookmark or alert in the document..

## Comment

## Preferred Paper Size

## Conditional Show

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

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

If this check box is marked, the report may be dynamically shown or hidden depending on the value of a condition expression entered in the edit box to the right. The condition expression will be evaluated each time the list of available reports is generated. The report will only be available when the condition returns true. Users with ADMIN privileges for the document can override all show conditions with the Show All Sheets and Sheet Objects in the dialog Document Properties: Security (page 423). 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 100) 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 101) 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:

| Initial Selections | These radio buttons allow you to state the initial selections for the report print. <br> Regardless of which setting you use, the selections prevailing before the report <br> print will be re-established after the print. |
| :--- | :--- |
| Current Selections | Use current selections as the basis of the report print (default). |
| Clear All | All current selections in the document will be cleared before printing the <br> report. After completed printing, the original selections will be re-applied. |
| Current Selections | A bookmark, which can be selected in the drop-down box will be applied <br> before printing the report. After completed printing, the original selections will <br> be re-applied. |
| Loop Report Over Pos- |  |
| sible Values in Field | When this check box is marked, the entire report will be printed repeatedly <br> while selecting each possible value in the field specified in the drop-down <br> box below. If there are no possible values in the specified field, nothing will <br> 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:

| Orientation | Each page can have its separate setting for orientation. Choose between Por- <br> trait and Landscape. The choice will be reflected in the page list pane above. |
| :--- | :--- |
| Page Type | Set if the report should be printed as a Single Page or as a Multi Page (for <br> long tables). |
| Adjust Print for Paper | This setting is only available for multi paper pages. <br> Use Paper as Needed |
| No scaling of the sheet object printout. As many pages as needed in each direc- <br> tion will be used. |  |
| Scale to $\mathbf{x} \%$ <br> The sheet object printout will be scaled to a fixed percentage of its original <br> size. As many pages as needed in each direction will be used. <br> Fit to $\mathbf{x}$ by $\mathbf{y}$ <br> The sheet object printout will be scaled to fit on a fixed number of pages. |  |

## Settings



## Page Settings, General tab

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

## Use Introduction and Appendix

Introduction Placement

Introduction Offset

## Appendix Placement

## Appendix Height

Appendix Offset

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 loose backward compatibility with QlikView versions prior to 7.5.

By marking one, two or three of the check boxes you can choose to print the introduction area on the First Page, on all Intermediate Pages and/or on the Last Page.

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.

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).

By marking one, two or three of the check boxes you can choose to print the appendix area on the First Page, on all Intermediate Pages and/or on the Last Page.

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.

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 144)) 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.

| Object ID | The sheet object ID of the sheet object to be printed. |
| :--- | :--- |
| Properties... | Opens the Properties dialog for the sheet object associated with the print item. <br> Changes made in this dialog will be applied on the actual sheet object and <br> reflected in the report print result. |
| Stretch | This control is only available for objects on single paper pages. Here you may <br> select how the print object should be sized to fit inside the rectangle assigned <br> to it. |
| Clip | Do not size the print object. If it is too large for the placeholder rectangle it <br> will be truncated. If it is too small there will be blank space around it. |
| Fill | Size the print object so that it fills the placeholder rectangle. |
| Fill with Aspect | Size the print object so that it fits in the placeholder rectangle while keeping <br> the original sheet object aspect ratio. |
| Use Border | The print object is normally printed on the page without any frame or border. <br> By selecting this check box, you may have a border around the object. |
| Frame Color | Press this button to select the border color. |
| Frame Width | Here you 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.

| Left | Sets the position of the currently selected item(s) relative to the left margin. 0 <br> is to the far left of the available print area. |
| :--- | :--- |
| Top | Sets the position of the currently selected item(s) relative to the top. 0 is at the <br> top of the available print are. |
| Width | Sets the size of the currently selected item(s) in $1 / 1000$ parts of the width of <br> the available print area. |
| Height | Sets the size of the currently selected item(s) in $1 / 1000$ parts of the height of <br> the available print area. |
| Z Level | Sets the layer of the currently selected item(s). If items overlap the item at the <br> lowest level (1) will be printed first, then the next layer will be printed and so <br> on. | 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 150) 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 154) 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 873) 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 873). |
| 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
Description
Select or deselect the check box in order to enable/disable the alert.
Add
Remove

## 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.
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 doc-
ument. 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 com-
mand) 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 doc-
ument 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 spec-
ifying 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 bal- <br> loon when the alert is fired. |
| :--- | :--- |
| Use Custom Popup | The Use Custom Pop-up group lets you define a custom pop-up format for the <br> selected alert. By checking this alternative, you can replace the default Alert |
| Pop-up Settings as defined in Document Properties: General (page 408). The |  |
| Pop-up Settings button opens the Pop-Up Window Settings (page 411) |  |
| dialog. |  |


| Mode | In the Mode group you can check the relevant options in order to define an <br> alert as an Interactive (automatic triggers in layout) and/or as an alert relevant <br> for external programs running QlikView in Batch mode (command line <br> execution), e.g. QlikView Publisher, via the special Automation API for Batch <br> triggers. If none of the check boxes in this group are marked, the alert can still <br> be manually checked via macros. |
| :--- | :--- |
| Description | A commentary field where the creator of an alert can describe the purpose of <br> an alert. It is only used in the list of alerts in this dialog. |
| Message | The Message edit box is where you type the message to be displayed with the <br> alert. For pop-up alerts the text will appear in the pop-up, for e-mail alerts this <br> is the body text of the e-mail. The message text may be defined as a Cal- |
| culated Formula (page 851) for dynamic update. Click the ... button to open |  |
| the Edit Expression dialog for easier editing of long formulas. |  |

### 17.5 Alert Wizard

The alert wizard helps with the task of defining an alert.
The process has the following basic steps:

| Step 1 - Name/Describe the Alert | page 156 |
| :--- | :--- |
| Step 2 - Define the Alert Condition | page 156 |
| Step 3 - Define when to Test the Alert Condition | page 157 |
| Step 4 - Define Test Delay | page 158 |
| Step 5 - Define the Action(s) to Take when the Alert Fires | page 159 |

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.

| What is an Alert? |  |  |  | - |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| An alert is a composite entity typically consisting of four basic parts: <br> 1) A condition, i.e. a QlikView expression forming a logical condition, that can be either true or false. <br> 2) A logical state (bookmark, clear all or current selection state) that should be applied before checking the state of the condition expression. <br> 3) One or more events which should lead to checking the alert. <br> 4) 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. <br> When an alert is checked, the condition is met and the actions are performed, it can be said that the alert has fired' <br> Click Next to continue. |  |  |  |  |  |
| $\square$ Dont show this page again |  |  |  |  |  |
|  |  |  |  |  |  |

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



[^4]In this step you define the alert condition and the selection state that should be used when checking the alert condition.
\(\left.$$
\begin{array}{ll}\text { Alert condition } & \begin{array}{l}\text { Enter a QlikView expression that should evaluate to true (non zero) when the } \\
\text { alert is to fire. }\end{array} \\
\text { Selection state when } \\
\text { testing alert }\end{array}
$$ \quad \begin{array}{l}Here you decide which selection state should be used when testing the result <br>
of the alert condition. There are three choices: <br>

Current Selection\end{array}\right\}\)| The alert condition will be tested using whatever selections are made at the |
| :--- |
| time of checking the alert. |
| All Clear |
| All selections will be temporarily cleared before testing the alert condition. |
| Bookmark |
| A specified bookmark will temporarily be applied before testing the alert con- |
| dition. |

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 interactively

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



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



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 bal- <br> loon when the alert is fired. |
| :--- | :--- |
| Send mail | Enable this option to send mail as a result of the alert being fired. The body of <br> 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 <br> may be defined as a QlikView expression. |
| Mail recipients | Enter a semi-colon separated list of mail addresses. Each of these addresses <br> will receive an e-mail message whenever the alert fires. The list may be <br> 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 279) 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 respec- <br> tive values are displayed. |
| :--- | :--- |
| Definition | Highlight a variable in the list above to display its definition. The text may be <br> freely edited. The changes will immediately be reflected in the Value column <br> 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 <br> 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 <br> should appear in the Expression Overview dialog. The Expression column <br> cannot be turned off. |
| :--- | :--- |
| Find/Replace... | Opens the Find/Replace dialog where you may search for and replace spe- <br> cific text within the current list of expressions. It is also possible to make <br> 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 com-
mand changes made so far will remain even if you use Cancel to leave the
dialog.

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

| Sheet ID | The sheet ID of the sheet where the expression occurs. Empty for document <br> expressions. |
| :--- | :--- |
| Sheet Name | The name (tab title) of the sheet where the expression occurs. Empty for doc- <br> ument expressions. |
| Object ID | The sheet object ID of the sheet object where the expression occurs. Empty for <br> document and sheet expressions. |
| Object Name | The name of the object where the expression occurs. |
| Location | The location of the expression within properties with 'friendly name', i.e. a sim- <br> ple explanation of the type of expression. |
| Label | The label of the main expression of the object. |
| Location (Full) | The exact location of the expression within the properties structure. The nota- <br> tion corresponds to the object property hierarchies of the QlikView Auto- <br> mation interface, which is explained in detail in the QlikView API Guide. |
| Expression | 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 165).
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 165).

### 20.2 Internal Files in QlikView Documents

The info prefix to Load (page 246) and Select (SQL) (page 262) 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, accesses via the following paths:

> qmem: $/ /<$ bundled $>/$ sounds/qv_bttn.wav
> qmem: $/ /<$ bundled $>/$ sounds/qv_can.wav
> qmem: $/ /<$ bundled $>/$ sounds/qv_clear.wav
> qmem: $/ /<$ bundled $>/$ sounds/qv_load.wav
> qmem://<bundled>/sounds/qv_lock.wav
> qmem://<bundled>/sounds/qv_menu.wav
> qmem://<bundled>/sounds/qv_ok.wav
> qmem://<bundled>/sounds/qv_redo.wav
> qmem://<bundled>/sounds/qv_reex.wav
> qmem://<bundled>/sounds/qv_save.wav
> qmem://<bundled>/sounds/qv_scrpt.wav
> qmem://<bundled>/sounds/qv_ulock.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 173)), 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 169) 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 181), accessed from the File menu. Changes to the data structure can be achieved by Renaming Fields (page 369) 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 51)). 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 393)) 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 246) 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 195) for more information.
In load statements, the full set of Script Expressions (page 283) 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 168).

### 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 173) by clicking the Connect button. This generates a Connect (page 227) statement in the script.
After that, you define the fields and tables to be loaded in a Select (SQL) (page 262) statement. To create this statement, click the Select button. See Create Select Statement (page 187) for more information.

## 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 246), Select (SQL) (page 262) or Binary (page 223) statements. The only way of changing data in a field (apart from Input Fields (page 170)) 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 457).

## 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 269) and Untag Field (page 271).

- \$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 429) 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 429) 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 429) the settings in the script will prevail.

### 22.2 Input Fields

Previously it was not possible to alter data in Fields (page 169) 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 241) 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 may contain a Special Input Field Aggregation Functions (page 797) and still be open for input. The change will then be distributed back to the underlying field values, based on predefined algorithms, such as e.g. "spread equally" or "spread proportionally".
Version 9 introduced an extension to the input fields, namely the possibility to specify relative change. The following syntax applies ( $n$ is a number):

| \%+n | increases the current value by $\mathrm{n} \%$ |
| :--- | :--- |
| \%-n | decreases the current value by $\mathrm{n} \%$ |
| $+=\mathbf{n}$ | increases the current value by n |
| $\mathbf{- = \mathbf { n }}$ | decreases the current value by n |
| \%=n | multiplies the current value with n |
| $/=\mathbf{n}$ | divides the current value with n |

## Examples:

$\mathbf{\%}+\mathbf{1 0}$ increases the current value with $10 \%$.
$+=56$ increases the current value with 56 .

* $=\mathbf{2}$ multiplies the value with 2 .
$1=\mathbf{2}$ 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 519) 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 283).
For more information about chart expressions see chapter Chart Expressions (page 765).
For more information about the use of expressions in other layout objects see chapter Calculated Formulas (page 851).

## 23 Edit Script Dialog



The Edit Script dialog
The Edit Script dialog is opened from the File Menu (page 53) 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 283).
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

\(\left.$$
\begin{array}{ll}\text { Reload } & \begin{array}{l}\text { Executes the script, closes the dialog and opens the Fields page in the } \\
\text { Sheet Properties dialog. This command is also available on the Edit } \\
\text { Script dialog toolbar. }\end{array} \\
\text { Debug... } & \begin{array}{l}\text { Starts the script execution in the debugger. The debugger is used for } \\
\text { searching for errors in the script. Every script statement can be monitored } \\
\text { and the values of the variables can be examined while the script is } \\
\text { executed. This command is also available on the Edit Script dialog tool- } \\
\text { bar. }\end{array} \\
\text { Edit Hidden Script... } & \begin{array}{l}\text { Opens a script tab for hidden script. Here it is possible to define a hidden } \\
\text { part of the script, which is executed before the general script at each } \\
\text { reload. Hidden script tabs are always shown to the left of the open script } \\
\text { tabs. A password is required to create or access hidden script. }\end{array} \\
\text { Change Hidden Script } & \begin{array}{l}\text { Opens the New Hidden Script Password dialog where a new password } \\
\text { for access to the hidden script can be set. }\end{array} \\
\text { Password... } & \begin{array}{l}\text { Opens the New Hidden Script Password dialog, where a password for }\end{array}
$$ <br>
Create Hidden Script <br>

The hidden script must be set, and then opens a new script tab in the\end{array}\right\}\)| script window called Hidden Script. Here it is possible to define a hid- |
| :--- |
| den 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. |

## Edit Menu

| Undo | Undoes the latest change (multiple step undo is possible). Equivalent to pressing CTRL+Z. |
| :---: | :---: |
| Redo | Redoes the latest Undo. Equivalent to pressing CTRL+Y. |
| Cut | Exports the selected text to the Clipboard. This command is is also available on the Edit Script dialog toolbar. Equivalent to pressing CTRL+X. |
| Copy | Copies the selected text to the Clipboard. This command is also available on the Edit Script dialog toolbar. Equivalent to pressing CTRL+C. |
| Paste | Pastes the contents of the Clipboard into the dialog at the position of the cursor. This command is also available on the Edit Script dialog toolbar. Equivalent to pressing CTRL+V. |
| Delete | Deletes the selected script text. Equivalent to pressing DEL. |
| Clear Entire Script | Clears the script, i.e. removes all the text except the initial autogenerated set statements. |
| Select All | Selects the entire script text. Equivalent to pressing CTRL+A. |
| Find/Replace... | Opens a dialog which makes it possible to find and replace numbers or characters in the script. This command is also available on the Edit Script dialog toolbar. Equivalent to pressing CTRL+F. |
| Goto... | Opens a dialog which makes it possible to go to a specified line number in the script. Equivalent to pressing CTRL+G. |
| Upper Case | Puts the selected script text in upper case. |
| Lower Case | Puts the selected script text in lower case. |
| Invert Case | Inverts the casing of the selected script text. |
| Capitalize | Capitalizes the selected script text. |
| Comment | Transforms a text row in the script to a comment. See also: Rem (page 257) |
| Uncomment | Transforms the text row back into original script text. |
| Indent | Indents the highlighted lines in the script. |
| Outdent | Outdents the highlighted lines in the script. |

## Insert Menu

## Set Statement

Environment Variables

Script File

Opens the Set Statement Wizard (page 179) where it is possible to enter a new set statement and choose a predefined statement.

Inserts the number interpretation variables in the script based on the operating system defaults.

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.

| Include Statement | Opens the Include Script Files dialog, where it is possible to browse for a <br> script file with one of the following formats: qus (a previously saved Qlik- <br> View script file), txt or sql. |
| :--- | :--- |
| Domain SID | Retrieves the NT domain security ID for use in the script. The NTDO- <br> MAINSID is employed in section access as one of the reserved fields that gov- <br> ern NT security. |
| Test Script | Inserts an autogenerated test script. |
| Load Statement | Inserts a load statement, either From File or Inline. Choosing Inline opens the <br> Inline Data Wizard (page 192) with the help of which it is possible to create <br> load inline statements from a spreadsheet style control. |
| Section Access | Inserts a section access statement either from a Pulisher Authorization table or |
| Inline. |  |
| Choosing Publisher Authorization opens the File Wizard: Options (page 207). |  |
| Connect Statement | Choosing Inline opens the Access Restriction Table Wizard (page 193). |
| Inserts a connect statement in the script. |  |

## Tab Menu

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

## Add Tab...

Insert Tab at Cursor...

Rename...
Promote

Demote
Merge with Previous...

Remove...

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.

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.

Opens a dialog for renaming the active tab.
Moves the active tab one step to the left. Tabs cannot be promoted to the left of the hidden script tab.

Moves the active tab one step to the right.
All text on the active tab is moved to the end of the preceding tab and the active tab is deleted.

Removes the active tab. The last remaining script tab cannot be removed.

## Tools Menu

ODBC Administrator 64 bit...

ODBC Administrator 32 bit...

Editor Preferences

Opens the ODBC Data Source Administrator for 64-bit ODBC drivers.

Opens the ODBC Data Source Administrator for 32-bit ODBC drivers.

Opens the User Preferences: Editor (page 82) page where it is possible to set the font and color of the different text types appearing in the script.

## 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.

OLE DB The connection to the data source is established using OLE DB.
ODBC The connection to the data source is established using ODBC.
QVSAdminDataProvider.dll This custom connector gives the possibility to connect to the QlikView Server and load information from the DMS and collaboration objects.

Custom... 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.

Force 32 Bit
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 <br> selected, or the Connect to Data Source (page 186) dialog where an <br> ODBC data source can be selected. |
| :--- | :--- |
| Select... | Opens the Create Select Statement (page 187) 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 <br> paths in the script. Relative paths are normally required when a document is to <br> be moved between different computers. This setting is also available in the <br> User Preferences dialog. If it is changed there, it will also be changed in the <br> Edit Script and Edit Hidden Script dialogs. |
| :--- | :--- |
| Use FTP | Enable this setting to select files from an FTP server, when clicking Table <br> Files, QlikView Files or Include. |
| Table files... | Opens the Open Local Files (page 190) dialog listing table files. Selecting <br> one or several files and pressing OK will generate one or several load state- <br> ments. When executed, the script loads data from the corresponding files. If the <br> Use FTP option is checked (see above), select a server from the list (or enter <br> 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 state- <br> ment is generated in the first line of the script. This statement loads the data <br> from the QlikView file but the layout settings will not be used. |
| Web Files... | Opens the File Wizard: Source (page 195) dialog where it is possible to enter <br> a URL as a source for the database table. |
| Field Data | Opens the File Wizard: Source dialog where it is possible to load the con- <br> tents 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.Function Name This drop-down list contains standard QlikView script functions.


#### Abstract

Paste Select a function in the Function Name 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.

Paste Pastes the selected variable into the script.
Show System Variables If this check box is marked, the list in the Variables drop down will include the system variables.

## Settings Page

The Settings page contains security settings for the QlikView script.
Script Privileges Open Databases in Read and Write Mode
If this option is selected the mode is write qualifier in select statements will be enabled for use.

## Can Execute External Programs

If this option is selected execute statements may be used in the script.

## Settings Scramble Connect User Credentials

If this check box is marked, the USERID and PASSWORD in connect statements will be scrambled in the script.

### 23.1 Set Statement Wizard



The Set Statement wizard

## Current Set Statement Variable Name <br> Either enter a name of a new variable or edit the predefined variable chosen below. The predefined variable is inserted here when clicking Paste. <br> Variable Value <br> 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. <br> Predefined Set Statements <br> Variable Groups <br> Choose what kind of variable to use. <br> Variables <br> Choose the variable to use. <br> Predefined Values <br> Choose between the predefined values in the variable. <br> Paste <br> Click the button to move the predefined values up to Current Set Statement for editing.

### 23.2 Find/Replace (Script)

| Find |  |
| :--- | :---: |
| Find What | Find Next |
|  | Replace |
| Replace With | Replace All |
| $\square$ | Co Dp |
| $\square$ Case Sensitive | Cown |
| $\square$ Word Search | Hearch all tabs |

## 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.

| Find What | The text string to search for. |
| :--- | :--- |
| Replace With | The text to replace the search string with. |
| Find Next | Moves selection to the next occurrence of the search string. |
| Replace | Makes a replacement in the selected section. |
| Replace All | Makes replacements in all occurrences of the search string. |
| Case Sensitive | If this check box is marked, the text search will be case sensitive. |
| Word Search | If this check box is marked QlikView will only find occurrences of the search <br> string forming a whole word (delimited by spaces or other non-letter char- <br> acters). |
| Search All Tabs | If this check box is marked, the find/replace operation will be made over all <br> script tabs. |
| Up | 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.


[^5]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):
## Click on Table Caption

Point at Table Caption Moves the table in the layout. and Drag

Hover Over Table Cap- Shows pop-up info about table. tion

Click on Field in Table Highlights the table, the field, all tables containing that field and the connections between them.

## Hover over Field in Table

Right-click in a Table
Shows pop-up info about field.

Click on Preview to see a dialog of sample records from the loaded data.


## Point at Connector Moves the connector point in the layout. <br> Point or Connector Bend and Drag <br> Right-click on Con- <br> Moves the connector point back to automatic positioning. nector Point or Connector Bend <br> Click on Layout Back- Cancels all highlighting of tables, fields and connectors. ground

The following commands are available in the dialog toolbar and buttons (click here to expand a list):

| Export Image | Opens a file browser dialog making it possible to save the current table <br> view as a bitmap or png image to a file on disc. |
| :--- | :--- |
| Export Structure | By pressing this button it is possible to export the table structure of the <br> document to a set of text files. <br> These text files, one for the tables (filename.Tables.tab), one for the fields <br> (filename.Fields.tab) and one for mapping in between (filename.Ma- <br> ppings.tab) can easily be read back into QlikView for further analysis <br> with the full power of the QlikView logic. A dialog will appear where <br> the target folder for the export can be chosen. The default is to put the <br> files in the same folder as the QlikView document. |
| Print Image | Opens the Print dialog from which it is possible to print the current table <br> view. Equivalent to pressing CTRL+P. |
| Copy Image | Copies the current table view as an image to clipboard. Equivalent to <br> pressing CTRL+C. |
| Auto Layout | Re-arranges the tables in the current view. |
| Ziew | The zoom factor of the current view can be set in this drop-down box. |
| Choose between Internal Table View and Source Table View (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 36) 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 returnes 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 173) 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.

| SQL Server ODBC Data Source Test |  |
| :--- | :--- |
| Test Results |  |
| Microsoft SQL Native Client Version 09.00.4035 | A. |
| Running connectivity tests... <br> Attempting connection <br> Connetion established <br> Verfifing option settings <br> Disconnecting from server <br> TESTS COMPLETED SUCCESSFULLY! |  |
| OK |  |

## 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 173). 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.
Data Source Name of the current ODBC/OLE DB data source.
Database It is possible to choose between available databases via the drop-down list.
Owner The database owner is shown here. It is possible to choose between available owners via the drop-down list.

Connect By clicking this button, the Connect to Data Source (page 186) dialog opens.

## Driver

Support
Click this button to open the ODBC Driver Information or OLEDB Driver Information dialog, containing information about supported functionality in the driver currently used.

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.
Database Tables 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).

Fields This list shows all available fields in the selected table. Choose one or more fields from the list to be included in the select statement. Use $" * "$ in order to select all of the field names at once. The selection of fields will be reflected in the Script page of the third group (see description below). Fields can be sorted in Original Order from the database or in alphabetical Text Order by selection in the drop-down control.

Show Icon for Key
Fields
Tables

Views This check box must be selected if database views are to appear in the Database Tables list.

Synonyms This check box must be selected if database table synonyms are to appear in the Database Tables list.

System Tables This check box must be selected if database system tables are to appear in the Database Tables list.

Aliases This check box must be selected if database table aliases are to appear in the Database Tables 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 appear- <br> ing on a separate row in the script. |
| :--- | :--- |
| Row | Select this option to generate the entire select statement appearing on a single <br> row in the script. |
| Structured | Select this option to generate the select statement on structured rows in the <br> script. |
| Add | If this option is marked, the select statement will be generated with a pre- <br> ceding load statement using the select statement as input. All fields will be <br> listed in the load statement, even if * is used in the select statement. |
| 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 state- |  |
| ment 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 $\mathbf{i}$ 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 In the list select the blob that should be resized.
data
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 $\mathbf{O K}$ 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 $(C T R L+C)$ 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 173).
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.

FTP Server Address

## User Name

## Password

## Connect

## Passive Semantics

Anonymous Logon
File Name

Files of Type
Open

The first time connecting to the FTP server, the address has to be typed in here.

Unless Anonymous Logon is checked, the user name will be entered here.
Unless Anonymous Logon is checked, the password will be entered here.
Click this button to connect to the selected server. Selecting a server directly from the list, bypasses this procedure.

To connect through a firewall, a passive FTP may have to be used.
Lets the user log on without stating a user name and password.
The name(s) of the selected file(s) will appear here, when connected to the server.

Specify the desired file type here.
Clicking this button generates a load statement that will enter into the load script in the Edit Script Dialog (page 173).

### 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.

| Insert Column | Inserts a new blank column if one or more columns are selected. |
| :--- | :--- |
| Delete Column | Deletes the selected column(s). |
| Insert Row | Inserts a new blank row if one or more rows are selected. |
| Delete Row | 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 401).

The dialog contains the following commands:
Fields to Use The list contains all the possible security fields in a QlikView section access. Mark the check box for those you wish to include.

Basic User Access
Table
Press this button to mark ACCESS, USERID and PASSWORD while unmarking the other fields.

Basic NT Security
Press this button to mark ACCESS, PASSWORD, NTNAME and NTDOMAINSID while unmarking the other fields.

When closing the dialog with OK, the Inline Data Wizard (page 192) 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 261) 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 173). 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.

| Local File | Enter the path to a local file or click Browse to find it through the explorer. |
| :--- | :--- |
| Internet File | Enter the path to the Internet file. |
| From Field | Opens the Document Data Source (page 196) dialog where it is possible to choose <br> 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.

## Field Value Tables

A drop-down menu showing the tables that are loaded in the active document.
Fields
The fields of the selected table.
Values
The values of the selected fields.
File Path The name of the selected table and field.
Value Type $\quad$ The type of the selected value.
Text
If the value is text it is displayed here.

## Info

If information is connected to the field value it is displayed here.

## Blob

If a blob is connected to the value it is displayed here.

## 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.
$\left.\begin{array}{ll}\begin{array}{l}\text { Header } \\ \text { Size }\end{array} & \begin{array}{l}\text { The header part of the file is a specified number of lines or bytes (only for delimited, fixed } \\ \text { record and Excel files). The header may be omitted. }\end{array} \\ \text { Character } & \begin{array}{l}\text { The character set of the table file is set here. If the desired character set is not available in } \\ \text { the drop-down menu, it is possible to choose Custom. This opens the Codepage selector } \\ \text { dialog, where to choose among the character sets installed on the computer. }\end{array} \\ \text { Set } & \text { Quoting } \\ \text { Comment the Quotes (page 252) scheme to be used. Choose between Standard, None and MSQ. }\end{array} \quad \begin{array}{l}\text { In some files, comments are used between records. Often a comment line begins with a spe- } \\ \text { cial character or a set of characters, e. g. //. Here it is possible to specify the character(s) that } \\ \text { mark the beginning of a comment to enable QlikView to recognize comments. }\end{array}\right\}$

## File Wizard: Type - Fixed Record

| Header Size |  |  | Tab Size |  |
| :---: | :---: | :---: | :---: | :---: |
| None $V$ | 0 | $\stackrel{1}{2}$ |  | $\hat{v}$ |
| Record Size |  |  |  | Embedded Labels... |
| None $\quad \checkmark$ | 0 | $\stackrel{\text { v }}{ }$ |  |  |
| Char Set |  |  |  | Analyze Fix Positions |
| Western European |  | $\checkmark$ |  | Clear Fix Positions |
| $\square$ Ignore EOF |  |  |  |  |

The File Wizard - Type page, fixed record file

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).

Record Size Specify the size of a record to be a specified number of lines or a specified number of bytes (only for fixed record files).

Char Set The character set of the table file is set here. If the desired character set is not available in the drop-down menu, choose Custom. This opens the Codepage selector dialog, where it is possible to choose among the character sets installed on the computer.

Ignore EOF 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.

Tab Size
The length of a tab set in spaces.
Embedded
Labels...
Embeds the labels. Should be used if the field names are stored in the first line of the table.

```
Analyze Fix Posi- Analyzes and sets the columns widths for fix record files and for some HTML
tions
Clear Fix Posi- Clears all column breaks (for fix record files and for some HTML tables).
tions
```


## File Wizard: Type - DIF

| Labels | Char Set |
| :--- | :--- | :--- |
| None | Western European |

The File Wizard - Type page, DIF file

| Labels | If the field names (column headers) are stored in the first line of the table, this con- <br> trol should be set to Embedded. The DIF format also allows field names to be <br> explicitly specified, and if that is the case, this control should be set to Explicit. If <br> the table contains no field names, the None alternative should be used. |
| :--- | :--- |
| Character Set | The character set of the table file is set here. If the desired character set is not avail- <br> able in the drop-down menu, choose Custom. This opens the Codepage selector <br> dialog, where it is possible to choose among the character sets installed on the com- <br> puter. |

## 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.

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 <br> containing several tables, e.g. HTML or Excel files. In Excel, all sheets in a work- <br> book and all named areas (e.g. multiple cells with names) in the worksheets are iden- <br> tified as possible tables. |
| :--- | :--- |
| Header Size | Here it is possible to omit the header part of the file: a specified number of lines or <br> 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 con- <br> trol should be set to Embedded. The DIF format also allows field names to be <br> explicitly specified, and if that is the case, this control should be set to Explicit. If <br> 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

Char Set
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.

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.

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 mininum 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 QlikView. 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
Conditional Delete...

Deletes the highlighted rows and columns.
Opens the Specify Row Condition (page 208) 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 209) 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 210) 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 <br> </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 211) 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 212) 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.

$$
\begin{array}{ll}
\text { Unwrap } & \begin{array}{l}
\text { Unwraps the table. The border between the two halves must be set with the cursor } \\
\text { first. It is possible to split the table either vertically or horizontally. }
\end{array} \\
\text { Conditional } & \begin{array}{l}
\text { Defines the condition for a vertical split of the table. Opens the Specify Row Con- } \\
\text { dition (page 208) dialog. }
\end{array}
\end{array}
$$

## 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.
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 Parameters


#### Abstract

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 212) dialog, with which is possible to create a <br> 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 214) 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.
Compare with Matches the cell with a cell condition that can evaluate to true or false.
value
Compare with col- Matches the cell with the corresponding cell in another column.
umn

Range Selects x rows, then skips y rows. Starting and ending positions for this may be specified with indexes.
\(\left.$$
\begin{array}{ll}\text { All Rows } & \text { Use this condition if all rows should be selected. } \\
\text { Column } & \text { Sets the column number on which the condition should be applied. } \\
\text { Orom } & \begin{array}{l}\text { Only visible in Range mode. Sets the row number from which rows should be } \\
\text { deleted or a second condition should be applied. }\end{array} \\
\text { Only visible in Range mode. Sets the row number to which rows should be deleted } \\
\text { or a second condition should be applied. }\end{array}
$$ \quad \begin{array}{l}Only visible in Range mode. Here it is possible to set whether rows should be <br>
selected or skipped cyclically, e.g. cyclically select two rows, then skip one. If all <br>

rows should be used, Select should be set to 1 and Skip should be set to 0 .\end{array}\right]\)| 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. |

## Fill Cells



The Fill Cells dialog

## Specification

Target Column

## 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

| Cell Condition |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- |
| Cell Value | is empty |  |  |  |
| Options |  |  |  |  |
| $\square$ Not | $\square$ Case Sensitive |  |  |  |
|  |  | $0 K$ | Cancel | Help |

## The Cell Condition dialog

This dialog opens from the Fill Cells (page 209) 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 locigal 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.

```
Cells and Col- Source Column
umns Sets the column number from which cell values should be copied.
    Cells from these rows
    Opens the Specify Row Condition (page 208) dialog where it is possible to set the
    row condition.
    Target Column
    Specifies the number of the target columns.
```


## Advanced Options Replace Null Value with Previous Value

```
Fills empty cells with the value in the cell above.
```


## Delete Source Row

```
Deletes the source row after copying its content.
```


## 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 $\langle\mathrm{BR}\rangle$ symbol.

| Cells to Insert | Number of cells to insert. Normally this should be set to 1, but if extra rows <br> 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 $\mathbf{O K}$, 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.
$\begin{array}{ll}\text { Advanced } & \text { Mark Advanced to type the clause from scratch. } \\ \text { Empty Template } & \text { This option will insert Where (1=1) into the script and it is possible to edit it fur- }\end{array}$ ther there.

## Crosstable

A cross table is a common type of table featuring a matrix of values between two orthogonal lists of header data. To turn a cross table into a straight table, use a crosstable prefix.
The syntax is:

```
crosstable (attribute field , data field [ , n ] ) ( loadstatement
| selectstatement )
```

where:
attribute field is the field to contain the attribute values.
data field is the field to contain the data values.
$n$ is the number of qualifier fields preceding the table to be transformed to generic form. Default is 1 .

## Examples:

Crosstable (Month, Sales) Load * from exl.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 syntax. 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

Data Field

The name of the new field that will contain all the fields (attribute values) to be transformed.

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. $\backslash$. Optional parameter. If omitted '/' will be used. |

```
Belongs To Param- Ancestor ID (AncestorID)
eters
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 |  |  |  |  |  |  |  | $x$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Max Line Length |  |  |  |  |  |  |  |  |
| 1 全 |  |  |  |  |  |  |  |  |
| ```LOAD Author, Title, Genre FROM Books.xlsx (ooxml, embedded labels, table is Sheet1);``` |  |  |  |  |  |  |  |  |

## 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 Lengh 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 171) 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.
Debug
This group contains settings that determine how the script is run in the
Debugger.
Run
This is the normal script execution. The script will progress to the end or until
a breakpoint is encountered.
Animate
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.
Step

This executes one script statement at a time. $\quad$| Enter a number in the spinner box below. The number entered is the maximum |
| :--- |
| number of accepted records at each load and select statement. This is a very |
| useful way to limit the execution time when the script is tested with live data. |
| Help |
| End Here |
| Opens the QlikView help. |
| Cancel |
| Buffers |

## 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 BackusNaur Formalism (page 905).

### 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:

| Add | page 222 |
| :--- | :--- |
| Alias | page 223 |
| Binary | page 223 |
| Buffer | page 223 |
| Bundle | page 224 |
| Call | page 225 |
| Comment Field | page 226 |
| Comment Table | page 226 |
| Concatenate | page 227 |
| Connect | page 227 |
| Crosstable | page 229 |
| Directory | page 230 |
| Disconnect | page 230 |
| Do..loop | page 231 |
| Drop Field | page 231 |
| Drop Table | page 232 |


| Execute | page 232 |
| :---: | :---: |
| Exit Script | page 233 |
| First | page 233 |
| For..next | page 233 |
| For each..next | page 234 |
| Force | page 236 |
| Generic | page 236 |
| Hierarchy | page 236 |
| HierarchyBelongsTo | page 237 |
| If..then..elseif..else..end if | page 238 |
| Image_size | page 239 |
| Info | page 239 |
| Inner | page 239 |
| Inputfield | page 241 |
| IntervalMatch | page 241 |
| IntervalMatch (Extended Syntax) | page 243 |
| Join | page 243 |
| Keep | page 244 |
| Left | page 244 |
| Let | page 246 |
| Load | page 246 |
| Loosen Table | page 253 |
| Mapping | page 253 |
| Map ... using | page 254 |
| NoConcatenate | page 254 |
| NullAsNull | page 255 |
| NullAsValue | page 255 |
| Outer | page 255 |
| Qualify | page 256 |
| Rem | page 257 |
| Rename Field | page 258 |
| Rename Table | page 258 |


| Replace | page 260 |
| :---: | :---: |
| Right | page 259 |
| Sample | page 261 |
| Section | page 261 |
| Select (SQL) | page 262 |
| Semantic | page 263 |
| Set | page 264 |
| Sleep | page 264 |
| SQL | page 264 |
| SQLColumns | page 265 |
| SQLTables | page 265 |
| SQLTypes | page 266 |
| Star | page 266 |
| Store | page 267 |
| Sub..end sub | page 267 |
| Switch..case..default..end switch | page 268 |
| Tag Field | page 269 |
| Trace | page 270 |
| Unless | page 270 |
| Unmap | page 271 |
| Unqualify | page 271 |
| Untag Field | page 271 |
| When | page 272 |

## 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:
If..then..elseif..else..end if
page 238
For..next
page 233

## For each..next

## Do..loop

Switch..case..default..end switch

## Exit Script

Sub..end sub
Call
page 234
page 231
page 268
page 233
page 267
page 225

## Add

The add prefix can be added to any Load (page 246), Select (SQL) (page 262) or Map ... using (page 254) statement in the script. It is only meaningful 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 add map...using statement causes mapping to take place also during partial script execution.
The syntax is:
add [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:
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 Tabl. 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 326)).
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 are strings representing what the respective names imply.

## Examples:

```
Alias ID_N as NameID;
Alias A as Name, B as Number, C as Date;
```

The name changes defined through this statement are used on all subsequent Select (SQL) (page 262) and Load (page 246) 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 access and data parts of a QlikView document. It does not load the layout information. Only one binary statement is allowed in the script and it can only be put as the first statement of a script. The syntax is:

## binary file

where:
file ::= [ path ] filename

## Examples:

```
Binary customer.qvw;
Binary c:\qv\customer.qvw;
```

The path is the path to the file, either absolute, or relative to the.$q$ vw 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 246) and Select (SQL) (page 262) statements in script. It indicates that QVD Files (page 393) 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 94). 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 housekeeping can be turned off by checking the options Keep Unreferenced QVD Buffers in Document Properties: General (page 408) (first requirement) and Keep Orphaned QVD Buffers in User Preferences (page 77) (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 select * from MyTable;
buffer (stale after 7 days) select * from MyTable;
buffer (incremental) load * from MyLog.log;
```

Back to Script Statements and Keywords (page 219).

## Bundle

If a piece of external information, such as an image or a sound file is to be connected (see Linking Information to Field Values (page 385)) to a field value, this is done in a table that is loaded with the Info (page 239) prefix. In order to maintain portability, it is possible to include the external files into the .qve 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 * from flagoecd.csv;
Bundle Select * from infotable;
```

Back to Script Statements and Keywords (page 219).

## Call

The call control statement calls a subroutine which must be defined by a sub statement. The syntax is:

## callname ( [ paramlist ])

Where:
name $\quad$ 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 field name, 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 + I
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 is 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 253)).

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 $:=(t a b l e\{, t a b l e\})$
mapname is the name of a mapping table previously read in a mapping load or mapping select statement (see Mapping (page 253)).

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 169), 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.
The syntax is:
concatenate [ (tablename ) ] ( loadstatement | selectstatement )

## Examples:

```
Concatenate load * from file2.csv;
Concatenate select * from table3;
tab1:
Load * from file1.csv;
tab2:
load * from file2.csv;
Concatenate (tab1) load * from file3.csv;
```

Back to Script Statements and Keywords (page 219).

## Connect

A general database can be accessed through the OLE DB/ODBC interface, but first, the data source needs to be specified. This is done 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 [ ( access_info ) ]
```

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 (page 281).
datasourcename must be a defined $O D B C$ data source.
conn-spec-item $::=\mathbf{D B Q}=$ database_specifier $\mid$ DriverID=driver_specifier $\mid \mathbf{U I D}=$ userid $\mid \mathbf{P W D}=$ password
The possible connection specification items may differ between different databases. For some databases, also other items than the above are possible. Normally only the database specifier is used.
access_info $::=$ access_item \{, access_item \}
access_item $::=$ (userid is userid $\mid$ xuserid isscrambledusername ) |(password is password $\mid$ xpassword isscrambledpassword) $\mid$ codepage is codepageID $\mid$ 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 77). 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.
The codepage is specifier can be used if problems is 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, Pass-
word is admin);
```

The data source defined through this statement is used by subsequent Select (SQL) (page 262) 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 has the same application 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 has the same application 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 orthogonal lists of header data. To turn a cross table into a straight table, use a crosstable prefix.
The syntax is:

```
crosstable (attribute field , data field [ , n ] ) ( loadstatement
| selectstatement )
```

where:
attribute field is the field to contain the attribute values.
data field is the field to contain the data values.
$n$ is the number of qualifier fields preceding the table to be transformed to generic form. Default is 1 .

## Examples:

```
Crosstable (Month, Sales) Load * from exl.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 syntax. 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 <br> be transformed. |
| Data Field | The name of the new field that will contain the data of the attribute values. |

## Directory

The Load (page 246) 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 246) 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 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 creates a loop which will be executed until a logical condition is met. The syntax is:

```
do[ ( while | until ) condition ] [statements]
[exit do [ ( when | unless ) condition ] [statements]
loop[ ( while | until ) condition ]
```

Where:
condition $\quad$ is a logical expression evaluating to true or false.
statements $\quad$ 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 fist 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-ofline, each of its three possible clauses (do, exit do and loop) must not cross a line boundary.

## Examples:

```
// load files file1.csv..file9.csv
for 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 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 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:

```
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-ofline, 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 246) or Select (SQL) (page 262) 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 expressing which evaluates to an integer indicating the maximum number of records to be read.

## Examples:

```
First 10 Load * from abc.csv;
First (1) Select * from Orders; (Note: The parenthesis is allowed but
not required.)
```

Back to Script Statements and Keywords (page 219).

## For..next

The for..next control statement creates a loop with a counter. The statements inside the loop enclosed by for and next will be executed for each value of the counter variable between specified low and high limits. The syntax is:

```
forcounter = expr1 to expr2 [ step expr3 ]
```

[statements]
[exit for [ (when | unless ) condition ]
[statements]
next[counter]

Where:
counter is a script variable name. If counter is specified after next it must be the same variable name as the one found after the corresponding for.
exprl is an expression which evaluates to the first value of the counter variable for which the loop should be executed.
expr 2 is an expression which evaluates to the last value of the counter variable for which the loop should be executed.
expr 3 is an expression which evaluates to 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, expr 2 and expr 3 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-ofline, 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 creates a loop witch executes for each value in a comma separated list. The statements inside the loop enclosed by for and next will be executed for each value of the list. Special syntax makes it possible to generate lists with file and directory names in the current directory. The syntax is:

```
for each var in list
```

[statements]
[exit for [ ( when | unless ) condition ]
[statements]

```
next[var]
```

Where:
$v a r$ 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.
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
end sub
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 246) and Select (SQL) (page 262) 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 375). The syntax is:
generic (loadstatement | selectstatement )

## Examples:

```
Generic Load * from abc.csv;
```

Generic Select * from table1;
Back to Script Statements and Keywords (page 219).

## Hierarchy

The hierarchy keyword is a prefix to the Load (page 246) and Select (SQL) (page 262) statements. It is used to transform an adjacent nodes table to an expanded nodes table. Hence, the internal table must be an adjacent nodes table. The output of the hierarchy prefix is an expanded nodes table with all the needed additional fields.
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 can still have any number of children. The table may of course contain additional fields describing attributes for the nodes. In expanded nodes tables 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. More details about adjacent nodes tables and expanded nodes tables can be found in the chapter Hierarchies in section Evaluating the Loaded Data (page 375).
Normally the internal table has exactly one record per node and in such a case the output table will contain the same number of records. However, sometimes there are nodes with multiple parents, i.e. one node is represented by several records. If so, the output table may have more records than the internal table.

All nodes without a parent id or with a parent id not found in the nodeID column will be considered as roots. Also, only nodes with a connection to a root node - direct or indirect - will be loaded, thus avoiding circular references.
Additional fields containing the name of the parent node, the path of the node and the depth of the node can be created.
The syntax is:

```
Hierarchy (NodeID, ParentID, NodeName, [ParentName], [PathSource],
[PathName], [PathDelimiter], [Depth]) (loadstatement | select-
statement)
```

where
NodeID is the name of the field that contains the node id.
ParentID is the name of the field that contains the node id of the parent node.
NodeName is the name of the field that contains the name of the node.
ParentName is a string used to name the new ParentName field. If omitted, this field will not be created.
PathSource is the name of the field that contains the name of the node used to build the node path.
Optional parameter. If omitted, NodeName will be used.
PathName is a string used to name the new Path field, which contains the path from the root to the node. Optional parameter. If omitted, this field will not be created.
PathDelimiter is a string used as delimiter in the new Path field. Optional parameter. If omitted, '/' will be used.
Depth is a string used to name the new Depth field, which contains the depth of the node in the hierarchy. Optional parameter. If omitted, this field will not be created.

## Example:

```
Hierarchy(NodeID, ParentID, NodeName) Load
NodeID,
ParentID,
NodeName,
Attribute
From data.xls (biff, embedded labels, table is [Sheet1$];
```

Back to Script Statements and Keywords (page 219).

## HierarchyBelongsTo

The hierarchybelongsto keyword is a prefix to the Load (page 246) and Select (SQL) (page 262) statements. The internal 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 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 375).
The prefix creates a table containing all ancestor-child relations of the hierarchy. The ancestor fields can then be used to select entire trees in the hierarchy. The output table usually contains several records per node.
An additional field containing the depth difference of the nodes can be created.
The syntax is:

```
HierarchyBelongsTo (NodeID, ParentID, NodeName, AncestorID, Ances-
torName, [DepthDiff])(loadstatement | selectstatement)
```

where
NodeID is the name of the field that contains the node id.
ParentID is the name of the field that contains the node id of the parent node.
NodeName is the name of the field that contains the name of the node.
AncestorID is a string used to name the new ancestor id field, which contains the id of the ancestor node.
AncestorName is a string used to name the new ancestor field, which contains the name of the ancestor node.
DepthDiff is a string used to name the new DepthDiff field, which contains the depth of the node in the hierarchy relative the ancestor node. Optional parameter. If omitted, this field will not be created.

## Example:

```
HierarchyBelongsTo (NodeID, ParentID, Node, Tree, ParentName) Load
NodeID,
ParentID,
NodeName
From data.xls (biff, embedded labels, table is [Sheet1$];
```

Back to Script Statements and Keywords (page 219).

## If..then..elseif..else..end if

The if..then control statement creates a conditional clause which makes 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 statement is used with Info (page 239) 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 multimedia presentation is to be Linking Information to Field Values (page 385) to a field value, this is done in a table that is loaded through 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 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 Select * from infotable;
Info Select Key, Picture From infotable;
```

Back to Script Statements and Keywords (page 219).

## Inner

The Join (page 243) and Keep (page 244) 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 with a full set of data from each.
If used before keep, it specifies that the 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:
select * from table1;
inner join select * from table2;
```


## QVTable

A B C
$1 \quad$ aa $\mathbf{x x}$

```
QVTab1:
select * from Table1;
QVTab2:
inner keep select * from Table2;
```


## QVTab1

A B

1 aa

## QVTab2

A $\quad \mathbf{C}$

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 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. In input fields all field values will be treated as distinct, regardless if several have the same value.
The syntax is:

## inputfieldfieldlist

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 to a Load (page 246) or Select (SQL) (page 262) statement is used for linking discrete numeric values to one or more numeric intervals. There is an advanced application of IntervalMatch involving an extended syntax, it is possible to read about IntervalMatch (Extended Syntax) (page 243). The general syntax is:

## intervalmatch (field) (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) while NULL limits extend the interval indefinitely (unlimited).
Before the IntervalMatch statement, the field containing the discrete data points (Time in the example below) must already have been read into QlikView. The IntervalMatch statement does not by itself read this field from the database table.
The intervals may be overlapping and the discrete values will be linked to all matching intervals.

## Example:

Look at the two tables below! The first one gives 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 function we will 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:

```
Select * from OrderLog;
Select * from Eventlog;
IntervalMatch ( Time ) select Start, End from OrderLog;
```

The following table box could 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 to a Load (page 246) or Select (SQL) (page 262) statement is used for matching discrete data to one or more dimensions that are changing over time. This is a very powerful and flexible feature that can be used for resolving with slowly changing dimensions by linking the specific key field(s) to the appropriate numeric intervals. 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 attributes that are to be matched using the interval.
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. Nonnumeric limits render the interval to be disregarded (undefined) while NULL limits extend the interval indefinitely (unlimited). The intervals may be overlapping and values will be linked by all matching intervals. In order to avoid undefined interval limits being disregarded, it is 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 255) statement.
Before the IntervalMatch statement, the field containing the discrete data points (Date in the example below) must already have been read into QlikView. The IntervalMatch statement does not by itself read this field from the database table.

## Example:

Inner Join IntervalMatch (Date, Key) Load FirstDate, LastDate, Key resident 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 logical table. The join is a natural join made over all the common Fields (page 169). The join statement may be preceded by one of the prefixes Inner (page 239), Outer (page 255), Left (page 244) or Right (page 259).
join [ (tablename ) ] (loadstatement | selectstatement )

## Examples:

```
Join Load * from abc.csv;
Join Select * from table1;
tab1:
Load * from file1.csv;
tab2:
```

```
load * from file2.csv;
join (tab1) load * from file3.csv;
```

Back to Script Statements and Keywords (page 219).

## Keep

The keep prefix between two Load (page 246) or Select (SQL) (page 262) statements has the effect of reducing one or both of the two tables before they are stored in QlikView, based on the intersection of table data. The keep keyword must always be preceded by one of the prefixes Inner (page 239), Left (page 244) or Right (page 259). 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.

## Note!

The explicit Join (page 243) keyword 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 tables instead of joining them, which greatly reduces memory usage, increases processing speed and offers enormous flexibility. Explicit joins should therefore be generally avoided in QlikView scripts. The keep functionality was designed to reduce the number of cases where explicit joins needs to be used.

Back to Script Statements and Keywords (page 219).

## Left

The Join (page 243) and Keep (page 244) 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 with a full set of data from 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 | $x x$ |
| :--- | :--- |
| 4 | $y y$ |

```
    QVTable:
    select * from table1;
    left join select * from table2;
```

A B C
1 aa $\mathbf{x x}$
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 264) statement, used for defining Script Variables (page 273). 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.

## Examples:

```
Set x=3+4;
Let y=3+4
z=$ (y) +1;
```

$\$(x)$ will be evaluated as ' $3+4$ '
$\$(\mathrm{y})$ will be evaluated as ${ }^{\prime} 7^{\prime}$
$\$(z)$ will be evaluated as ' 8 '

```
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 (page 169) from a typed ASCII file, from data defined in the script, from a previously loaded table, from the result of a following select statement or by generating data automatically. The general syntax of the load statement is:

```
load [ distinct ] *fieldlist
[( from file [ format-spec ] |
from_fieldfield [format-spec]
inline data [ format-spec ] |
residenttable-label |
autogenerate size )]
[ where criterion ] | while criterion ]
[ group_by fieldlist ]
[order_by field [ sortorder ] { , field [ sortorder ] } ]
```

where:
distinct is a predicate used if only the first of duplicate records should be loaded.
*fieldlist $::=(* \mid$ field $\{$, field $\})$
A list of the fields to be loaded. Using * as field list indicates all fields in the table.

## field $::=$ ( fieldref $\mid$ expression ) [ as aliasname ]

The field definition must always contain a reference to an existing field, or an expression.
fieldref $::=$ (fieldname $|@ f i e l d n u m b e r| @ s t a r t p o s: e n d p o s[\boldsymbol{I}|\boldsymbol{U}| \boldsymbol{R} \mid \boldsymbol{B}]$ )
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 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 $\mathbf{I}$ or $\mathbf{U}$, the bytes read will be interpreted as a binary signed (I) or unsigned ( $\mathbf{U}$ ) integer (Intel byte order). The number of positions read must be 1, 2 or 4. If @startpos:endpos is immediately followed by the character $\mathbf{R}$, 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 $\mathbf{B}$, 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 283).
as is used for assigning a new name to the field.
from is used if data should be loaded from a file, and not typed within the script.
file $::=$ [ path ] filename
The path is the path to the file, either absolute, or relative to the QlikView document (.qvw file). If the path is omitted, QlikView searches for the file in the directory specified by the directory statement. If there is no directory statement, QlikView searches in the working directory, which is usually the directory in which the QlikView file is located. The path may also be a URL address (HTTP or FTP), pointing to a location on the Internet or an intranet.
The filename may contain the standard DOS wildcard characters (* and?). This will cause all the matching files in the specified directory to be loaded.
format-spec $::=(f$ spec-item $\{, f$ spec-item $\})$
The format specification consists of a list of several Format Specification Items (page 250), within brackets.
from_field is used if data should be loaded from a previously loaded field.
field: $:=$ (tablename, fieldname)
The field is the name of the previously loaded tablename and fieldname.
format-spec $::=($ fspec-item $\{$, fspec-item $\})$
The format specification consists of a list of several Format Specification Items (page 250), 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 192) for help with the creation of load inline statements.
data $::=$ [ text ]
Data entered through an inline clause must be enclosed by Quotation Marks (page 281) 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.
format-spec $::=($ fspec-item $\{$, fspec-item $\})$
The format specification consists of a list of several Format Specification Items (page 250), within brackets.
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 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.
fieldname, path, filename and aliasname are text strings representing what the respective names imply.
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.
fieldlist $::=(f i e l d ~\{$, field $\}$ )
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.
field specifies which field 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).
sortorder is either asc for ascending or desc for descending. If no sortorder is specified, asc is assumed.

## Note!

If no source of data is given by means of a from, inline, resident or autogenerate clause, data will be loaded from the result of the immediately succeeding Select (SQL) (page 262) or load statement. The succeeding statement should not have a prefix.

## Examples:

Different file formats

```
Load * from datal.csv;
Load * from 'c:\userfiles\datal.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 datal.csv;
Load @1as A, @2 as B from data3.txt' (ansi, txt, delimiter is '\t', no
labels);
Load FirstName&' '&LastName as Name from datal.csv;
Load Quantity, Price, Quantity*Price as Value from data1.csv;
```


## Selecting certain records

```
Load distinct FirstName, LastName, Number from datal.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 tabl 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 $|\mathbf{o e m}| \mathbf{m a c} \mid$ UTF-8 | Unicode $\mid$ txt $\mid$ fix $\mid$ dif $\mid$ biff $|\mathbf{h t m l}| \mathbf{x m l} \mid$ qvd |delimiter is char $\mid$ no eof $\mid$ embedded labels $\mid$ explicit labels $\mid$ no labels $\mid$ table is [tablename $]$ header is $n \mid$ header is line $\mid$ header is $n$ lines $\mid$ comment is string $\mid$ record is $n \mid$ record is line $\mid$ record is $n$ lines $\mid$ no quotes $|\mathbf{m s q}|$ filters (filter specifiers )]
The following properties can be defined:

| Character Set | page 391 |
| :--- | :--- |
| Table Format | page 251 |
| Delimiter | page 251 |
| No eof | page 251 |
| Labels | page 251 |
| Header Size | page 252 |
| Record Length | page 252 |
| Quotes | page 252 |

## Character Set

Available character sets are:
ansi
oem
mac
UTF-8
Unicode
The file can be written with the ansi character set (Windows), with the oem character set (DOS, OS/2 and others), Unicode, UTF-8 or with the mac. The conversion from the oem character set is not implemented for MacOS. If nothing is specified, ansi is assumed under Windows.

## Example:

```
load * from a.txt (CodePage is 10000, txt, delimiter is ',', no
labels)
```

Back to Script Statements and Keywords (page 219).

## Table Format

| txt | In a delimited text file, .txt, the columns in the table are separated by some character. |
| :---: | :---: |
| fix | In a fixed record length file, fix, each column is exactly a certain number of characters wide. |
| dif | In a .dif file, (Data Interchange Format) a special format for defining the table is used. |
| biff | QlikView can also interpret data in standard Excel files by means of the biff format (Binary Interchange File Format). |
| html |  |
| qvd | The format qvd is the proprietary $Q V D$ Files (page 393) format, exported from a QlikView document. |
| qvx | qvx 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 ischar

For delimited table files, an arbitrary delimiter 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.
$" \ \backslash "$ representing a backslash ( $\backslash$ ) 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 often denotes end-of-file is disregarded.

Back to Script Statements and Keywords (page 219).

## Labels

## Labels (field names) <br> embedded labels

## explicit labels <br> 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.

Back to Script Statements and Keywords (page 219).

## Header Size

The syntax is:

## header is $n$

## header is line

## header isnlines

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 $\boldsymbol{n}$ ), or in lines (header is line or header is $\boldsymbol{n}$ lines). $n$ must be a positive integer, representing the header length. If not specified, header is $\mathbf{0}$ is assumed. The header is specifier is only relevant for Table file.

Back to Script Statements and Keywords (page 219).

## Record Length

Fixed record files only.

```
record is }
```


## record is line

## record isnlines

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

Text files only.

## no quotes

If Quotation Marks (page 281) are not to be accepted in a text file, the no quotes specifier should be used. By default, the quotes " " or ' ' can be used, but only if they are the first and last non blank character of a field value. The no quotes specifier is only relevant for txt files.

## msq

$\mathbf{m s q}$ indicates that modern style quoting is assumed.

Back to Script Statements and Keywords (page 219).

## XML (XML Files Only)

## xmlsax

## xmlsimple

## pattern ispath

Xmlsax and xmlsimple are mutually exclusive, only one can be specified when using xml. When using pattern, the file will be read from the start of the specified tag to the end of the tag. If path contains spaces it must be quoted.

## 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

One or more QlikView internal tables can be explicitly declared loosely coupled (see Logic and Selections (page 107) for details on loosely coupled tables) 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.

Back to Script Statements and Keywords (page 219).

## Mapping

The mapping prefix to a Load (page 246) or Select (SQL) (page 262) statement is used for mapping a certain field value or expression to the values in the first column of the mapping table. A mapping table consists of two columns, the fist containing comparison values and the second containing the desired mapping values. Mapping tables will be stored temporarily in the memory and dropped automatically, after script execution. The syntax is:
mapping( loadstatement |selectstatement )

## Examples:

```
mapping load * from x.csv
```

```
mapping 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 253) statement. 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 (see Mapping (page 253)).

## Examples:

```
map Country using Enables mapping of field Country using map Cmap.
Cmap;
\(\operatorname{map} \mathbf{A}, \mathbf{B}, \mathbf{C}\) using X; \(\quad\) Enables mapping of fields A, B and C using map X.
map * using GenMap; Enables mapping of all fields using GenMap.
```

The automatic 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 apply map function has to be used instead.
Back to Script Statements and Keywords (page 219).

## NoConcatenate

The NoConcatenate prefix forces two tables with identical field sets (i.e. they would normally be concatenated automatically) to be treated as two separate internal tables.
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

By default, QlikView considers NULL values to be missing or undefined entities. However, certain database contexts imply that NULL values are to be considered as special values rather than simply missing values. The fact that NULL values are normally not allowed to link to other NULL values can be suspended by means of the NullAsValue statement. Thus the NULL values of the specified fields are considered to be unknown rather than undefined when the NullAsValue statement is employed. The NullAsValue statement could be considered to supersede the NullDisplay (page 275) variable at field level.
The NullAsValue statement operates as a switch. It can be switched off again by means of the NullAsNull (page 255) statement.
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;
Load A,B from x.csv;
```

Please note the following points when using the NullAsValue statement:

- Linking of NULL values is always turned off by default at the beginning of script execution.
- Linking of NULL values of a specific field can be activated at any time using a NullAsValue statement.

Back to Script Statements and Keywords (page 219).

## NullAsNull

The NullAsNull statement turns off the linking of NULL values previously set by a NullAsValue (page 255) statement (follow this link for more information).
The syntax is:

## NullAsNull*fieldiist

*fieldlist is a comma separated list of the fields for which NullAsNull 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:

```
NullAsNull A,B;
Load A,B from x.csv;
```

Back to Script Statements and Keywords (page 219).

## Outer

The explicit Join (page 243) 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 outer keyword is optional.
outer join [ (tablename ) ] (loadstatement |selectstatement )

## Example:

## Table1

A B

1 aa

2 cc

3 ee

## Table2

A C

1 xx
$4 \quad$ yy

```
select * from table1;
join select * from table2;
```

OR
select * from table1;
outer join select * from table2;

## Joined table

A B C

1 aa $\mathbf{x x}$

2 cc -
3 ee -
$4 \quad-\quad$ y
Back to Script Statements and Keywords (page 219).

## Qualify

The automatic join between fields with the same name in different tables can be suspended by means of the qualify statement, which qualifies the field name with its table name. If qualified, the field name(s) will be renamed when found in a table. The new name will be in the form of tablename.fieldname. Tablename is equivalent to the label of the current table, or, if no label exists, to the name appearing after from in load and select statements.
Qualification is always turned off by default at the beginning of script execution. Qualification of a field name can be activated at any time using a qualify statement. Qualification can be turned off at any time using an Unqualify (page 271) 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.

## Note!

The qualify statement should not be used in conjunction with partial reload!

## Examples:

```
Qualify B;
Load A,B from x.csv;
Load A,B from y.csv;
```

The two tables x.csv and y.csv are joined only on $\mathbf{A}$. Three fields will result: A, x.B, y.B.

In an unfamiliar database, 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 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 */.
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).
The syntax is:

```
rename field [s] (using mapname | oldname to newname{ , oldname to
newname })
```

where:
mapname is the name of a previously loaded mapping table containing one or more pairs of old and new field names.
oldname is the old field name and
newname is the new field name.
Either syntax: rename field or rename fields can be used.

## Examples:

```
Rename field XAZ0007 to Sales;
FieldMap:
Mapping 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).
The syntax is:

```
rename table[s] (using mapname | oldname to newname{, oldname to
```

newname \})
where:
mapname is the name of a previously loaded mapping table containing one or more pairs of old and new table names.
oldname is the old table name and
newname is the new table name.
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 243) and Keep (page 244) 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 with a full set of data from 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 $\mathbf{x x}$
$4 \quad-\quad$ yy

QVTab1:
select * from Table1;
QVTab2:
right keep select * from Table2;

## QVTab1

## A B

1 aa

## QVTab2

A C

1 xx
$4 \quad y \mathbf{y}$
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 246), Select (SQL) (page 262) or Map ... using (page 254) 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 Filel.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 246) or Select (SQL) (page 262) statement is used for loading a random sample of records from the data source.
The syntax is:

```
samplep (loadstatement | selectstatement )
```

where:
$p$ is an arbitrary expression which valuates to a number larger than 0 and lower or equal to 1 . The number indicates the probability for a given record to be read.

## Examples:

```
Sample 0.15 Select * from Longtable;
Sample(0.15) Load * from Longtab.csv; (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 246) and Select (SQL) (page 262) statements should be considered as data or as a definition of the access rights. The syntax is:

## section(access | application)

## Examples:

```
Section access;
Section application;
```

If nothing is specified, section application is assumed. The section definition is valid until a new section statement is made.

Back to Script Statements and Keywords (page 219).

## Select (SQL)

The selection of Fields (page 169) from an ODBC data source is made through standard SQL SELECT statements. However, the extent to which the SELECT statements are accepted depends largely on the ODBC driver used. 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 counted once.
distinctrow is a predicate used if duplicate records in the source table only should be counted once.
*fieldlist $::=(* \mid$ 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 $\mid$ 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 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 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 specified in the data source definition of the data source in the Connect (page 227) 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.

Back to Script Statements and Keywords (page 219).

## Set

The set statement is used for defining Script Variables (page 273). These can be used for substituting strings, paths, drives, etc.
The syntax is:
setvariablename =string

## Examples:

```
Set FileToUse=Datal.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 to send an arbitrary SQL command through an $O D B C$ connection. Simply type:

## SQL

followed by the command.

## Example:

```
SQL leave;
```

Sending SQL statements which update the database will return an error, since QlikView per definition accesses all ODBC sources 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 262), 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 $O D B C$ data source, to which a connect has been made. These fields can be combined with the fields generated by the SQLTables (page 265) and SQLTypes (page 266) 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 $O D B C$ data source, to which a connect has been made. These fields can be combined with the fields generated by the SQLColumns (page 265) and SQLTypes (page 266) 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 $O D B C$ data source, to which a connect has been made. These fields can be combined with the fields generated by the SQLColumns (page 265) and SQLTables (page 265) 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 used for representing the set of all the values of a field in the database can be set through the star statement. It affects the subsequent Load (page 246) and Select (SQL) (page 262) statements. The syntax is:

```
star is[ string ]
```

where:
string is an arbitrary text. Note that the string must be enclosed by Quotation Marks (page 281) if it contains blanks.

## Examples:

```
Star is *;
Star is %;
Star is;
```

If nothing is specified, star is; is assumed, i.e. there is no star symbol available unless explicitly specified. This definition is valid until a new star statement is made.

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 246). The store statement to a CSV file does not support BIFF export.

```
store[ *fieldlist from] table into filename [format-spec ];
```

$*$ fieldlist $::={ }^{*} \mid$ field $)\{$, field $\left.\}\right)$ 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 (page 281) 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 $::=(\mathbf{t x t} \mid \mathbf{q v d}))$
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 $\quad$ is the name of the subroutine.
paramlist $\quad$ is a comma separated list of variable names for the formal parameters of the subroutine.
statements $\quad$ is any group of one or more QlikView script statements.

Arguments are copied into the subroutine and, if the corresponding argument in the calling 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.

## Examples:

```
// Example 1
sub INCR (I,J)
I = I + I
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}=\textrm{B}+
C=C+1
end sub
A=1
X=1
C=1
call ParTrans (A, (X+1)*2)
// A=2 (copied back from sub)
// X=1 (expression not copied back)
// C=1 (C in sub is local)
// B=NULL (local in sub)
```

Back to Script Statements and Keywords (page 219).

## Switch..case..default..end switch

The switch control statement creates a fork where one of several groups of statements is executed, depending on the value of an expression. The syntax is:

```
switch expression { case valuelist [ statements ]} [ default state-
ments lend switch
```

Where:

| expression | is an arbitrary expression. |
| :--- | :--- |
| valuelist | is a comma separated list of values with which the value of expression will <br> be compared. Execution of the script will continue with the statements in the <br> first group encountered with a value in valuelist equal to the value in expres- <br> sion. Each value in valuelist may be an arbitrary expression. If no match is <br> found in any case clause, the statements under the default clause, if specified, <br> will be executed. |
| statements | 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 253) load or Mapping (page 253) 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 string to the Script Execution Progress window and to the script log file, when used.
The syntax is:

## trace string

## Examples:

```
trace Main table loaded;
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 Mapping (page 253) of specified fields from the current point in the script. 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 256) 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 253) load or Mapping (page 253) 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 246) or Select (SQL) (page 262) 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 $\mathbf{v T o d a y}=$ Num(Today 0$)$; // returns the date serial number of today.

The following script variables are available:

Error Variables
Number Interpretation Variables
System Variables
Value Handling Variables
page 278
page 354
page 274
page 275

## 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 sys-tem-defined variable.

## \$(Include =filename )

The include variable specifies a file that contains text that should be included in the script. The entire script can thus be put in a file. A user-defined variable.

## Example:

```
$(Include=abc.txt);
```

I.e. the construction set Include =filename is not applicable.

## HidePrefix

All field names beginning with this text string will be hidden in the same manner as the system fields. A user-defined variable.

## Example:

set HidePrefix=' _' ;
If this statement is used, the field names beginning with an underscore will not be shown in the field name lists when the system fields are hidden.

## HideSuffix

All field names ending with this text string will be hidden in the same manner as the system fields. A userdefined 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 $O D B C$ 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)';
```


## 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';

## 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]';

## 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';
```

Back to Other Functions.

## Error Variables

The following variables can be used for error handling in the script:

| ErrorMode | page 278 |
| :--- | :--- |
| ScriptError | page 278 |
| ScriptErrorDetails | page 279 |
| ScriptErrorCount | page 279 |
| ScriptErrorList | page 279 |

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 $=\mathbf{0}$ QlikView will simply ignore the failure and continue script execution at the next script statement. By setting ErrorMode $\mathbf{= 2}$ 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:

1 No error
2 General Error

4 General ODBC Error
5 General OLE DB Error
6 General XML Error
7 General HTML Error
$8 \quad$ File Not Found
9 Database Not Found
10 Table Not Found
11 Field Not Found
12 File Has Wrong Format

## 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 syntax

## \$(variablename)

is used. \$(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)

is used. \$(variablename ) always yields a legal decimal-point 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.

## 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 $X=\$(\operatorname{MUL}(3,7)) ; / /$ 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 $\backslash$ MyScript.qvs);

### 24.5 Quotation Marks

If quotation marks are possible or necessary within a script statement, it is mentioned below, see also Script Statements and Keywords (page 219). In a Load (page 246) statement the following symbols should be used as quotation marks:

$$
\begin{array}{ll}
\text { Field names: } & {[], " ~ ", ~ ' ` o r ' '} \\
\text { String literals: } & \text { '' }
\end{array}
$$

For a Select (SQL) (page 262) statement interpreted by the ODBCdriver it may be slightly different, though. Usually the straight double quotation marks (ALT +0034 ) should be used for names and the straight single quotation marks (ALT +0039 ) for literals, and thus never the oblique: ' ' or ' '. However, some $O D B C$ drivers not only accept the oblique quotation marks, but also prefer them. In such a case the generated Select (SQL) (page 262) statements contain oblique quotation marks.

## Example: Difference Between Names and Literals

The difference between names and literals becomes clearer comparing the following expressions:

## 'Sweden' as Country

as a part of the field list in a load or select statement will load the text string "Sweden" as field value into the QlikView field "Country", whereas

## "land" as Country

as a part of the field list in a load or select statement will load the content of the database field or table column named "land" as field values into the QlikView field "Country". I.e. land will be treated as a field reference.

## Example: Difference Between Numbers and String Literals

The difference between names and string literals becomes clearer comparing the following expressions:

## '12/31/96'

as a part of an expression will be interpreted as the text string "12/31/96", whereas 12/31/96
as a part of an expression will be interpreted numerically as the date Dec. 31, 1996, and assigned the corresponding numerical value, provided that the system date format is MM/DD/YY.

## Example: MS Access

Microsoft Access $O D B C$ Driver 3.4 (included in MS Access 7.0) accepts the following quotes when analyzing the select statement:

Field names and table names: [ ] , " " or ` `
String literals:
Not accepted:

## 25 Script Expressions

Expressions can be used in both the Load (page 246) statement and the Select (SQL) (page 262) 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 169) and Operators (page 767), 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 767) 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 |
|  | operator1 expression |
|  | expression operator 2 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.
operatorl is a unary operator (working on one expression, the one to the right).
operator 2 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:

| Numeric Operators | page 768 |
| :--- | :--- |
| String Operators | page 768 |
| Logical Operators | page 768 |
| Relational Operators | page 768 |
| Bit Operators | page 769 |

## Numeric Operators

All numeric operators use the numeric values of the operands and return a numeric value as result.

```
+ Sign for positive number (unary operator) or arithmetic addition. The binary
``` operation returns the sum of the two operands.
- Sign for negative number (unary operator) or arithmetic subtraction. The unary operation returns the operand multiplied by -1 , and the binary the difference between the two operands.
* Arithmetic multiplication. The operation returns the product of the two operands.
/ Arithmetic division. The operation returns the ratio between the two operands.

\section*{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.
\&
like

String concatenation. The operation returns a text string, that consists of the two operand strings, one after another.

Example:
'abc' \& 'xyz' returns 'abcxyz'
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)

\section*{Logical Operators}

All logical operators interpret the operands logically and return true \((-1)\) or false ( 0 ) as result.
\begin{tabular}{ll} 
not & \begin{tabular}{l} 
Logical inverse. One of the few unary operators. The operation returns the log- \\
ical inverse of the operand.
\end{tabular} \\
and & Logical and. The operation returns the logical and of the operands. \\
or & Logical or. The operation returns the logical or of the operands. \\
Xor & \begin{tabular}{l} 
Logical exclusive or. The operation returns the logical exclusive or of the oper- \\
ands. I.e. like logical or, but with the difference that the result is false if both \\
operands are true.
\end{tabular}
\end{tabular}

\section*{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.
\begin{tabular}{|c|c|c|}
\hline < & 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. \\
\hline <= & 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. \\
\hline > & 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. \\
\hline >= & 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. \\
\hline = & 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. \\
\hline <> & 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. \\
\hline precedes & ASCII less than & \begin{tabular}{l}
Unlike the < 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. \\
Example: \\
' 11' precedes ' 2'returns true compare this to: \\
' 11 ' < ' 2'returns false
\end{tabular} \\
\hline
\end{tabular}

\section*{follows \\ ASCII greater than}

Unlike the > 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.

\section*{Example:}
' 23' follows ' 111'returns true compare this to:
' 23' > ' 111' returns false

\section*{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.
\begin{tabular}{lll} 
bitnot & Bit inverse. & \begin{tabular}{l} 
Unary operator. The operation returns the logical inverse of the operand per- \\
formed bit by bit.
\end{tabular} \\
bitand & Bit and. & The operation returns the logical AND of the operands performed bit by bit. \\
bitor & Bit or. & The operation returns the logical OR of the operands performed bit by bit. \\
bitxor & \begin{tabular}{l} 
Bit exclusive \\
or.
\end{tabular} & \begin{tabular}{l} 
The operation returns the logical exclusive or of the operands performed bit \\
by bit.
\end{tabular} \\
\(\gg\) & Bit right shift. & \begin{tabular}{l} 
Unary operator. The operation returns the operand shifted one step to the \\
right.
\end{tabular} \\
\(\ll\) & Bit left shift. & Unary operator. The operation returns the operand shifted one step to the left.
\end{tabular}

\subsection*{25.2 Aggregation Functions}

These functions can only be used in field lists for Load (page 246) statements with a group by clause. The aggregation functions are:
\begin{tabular}{ll} 
Basic Aggregation Functions & page 286 \\
String Aggregation Functions & page 288 \\
Counter Aggregation Functions & page 288 \\
Advanced Aggregation & page 289 \\
Statistical Aggregation Functions in Script & page 290 \\
Financial Aggregation Functions in Script & page 294 \\
Statistical Test Functions in Script & page 295
\end{tabular}

\section*{Note!}

If no value is found, NULL is returned for all Aggregation Functions, except Sum and Count which both return 0

\section*{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.

\section*{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.

\section*{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.

\section*{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;

```

\section*{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.

\section*{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.

\section*{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;

```

\section*{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 n th value in order will be returned. If the word distinct occurs before the expression, all duplicates will be disregarded.

\section*{Example:}
```

Load Customer,

```
```

firstsortedvalue(PurchasedArticle, OrderDate) as FirstProductBought
from abc.csv
group by Customer;

```

\section*{String Aggregation Functions}

\section*{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.

\section*{Example:}
```

Load Month, MinString(Month) as FirstSalesMonth from abc.csv group by
Year;

```

\section*{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.

\section*{Example:}
```

Load Month, MaxString(Month) as LastSalesMonth from abc.csv group by
Year;

```

\section*{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.

\section*{Example:}

Load City, FirstValue (Name), as FirstName from abc.csv group by City;

\section*{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.

\section*{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.

\section*{Example:}
```

Load Department, concat(Name,';') as NameList from abc.csv group by

```
Department;

\section*{Counter Aggregation Functions}

\section*{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.

\section*{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 per
Month;

```

\section*{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.

\section*{Example:}

Load Month, NumericCount(Item) as NumberOfNumericItems from abc.csv group by Month;

\section*{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.

\section*{Example:}
```

Load Month, TextCount(Item) as NumberOfTextItems from abc.csv group by

``` Month;

\section*{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.

\section*{Example:}

Load Month, NullCount (Item) as NumberOfNullItems from abc.csv group by Month;

\section*{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.

\section*{Example:}
```

Load Month, MissingCount(Item) as NumberOfMissingItems from abc.csv
group by Month;

```

\section*{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 799) expression.
By using this function in Add calculated dimension... (page 602) it is possible to achieve nested chart aggregation in multiple levels. See also Nested Aggregations and Related Issues (page 845).
When used in chart expressions it is possible to achieve Sum of Rows in Pivot Tables (page 847).

\section*{Examples:}
```

aggr( sum(Sales), Country )
aggr( nodistinct sum(Sales), Country )
aggr( sum(Sales), Country, Region )
count( aggr( sum(Sales), Country ))

```

\section*{Statistical Aggregation Functions in Script}

\section*{fractile (expression, fractile)}

Returns the fractile of expression over a number of records as defined by a group by clause.

\section*{Example:}

Load Class, fractile( Grade, 0.75 ) as \(F\) from abc.csv group by Class;

\section*{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.

\section*{Example:}
```

Load Month, kurtosis(Sales) as SalesKurtosis from abc.csv group by Month;

```

\section*{correl (x-expression, \(y\)-expression)}

Returns the aggregated correlation coefficient for a series of coordinates represented by paired numbers in xexpression 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.

\section*{Example:}
```

Load Month, correl(X,Y) as CC from abc.csv group by Month;

```

\section*{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.

\section*{Example:}

Load Month, avg(Sales) as AverageSalesPerMonth from abc.csv group by Month;

\section*{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.

\section*{Example:}
```

Load Month, stdev(Sales) as SalesStandardDeviation from abc.csv group
by Month;

```

\section*{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.

\section*{Example:}
```

Load Month, skew(Sales) as SalesSkew from abc.csv group by Month;

```

\section*{median (expression)}
returns the aggregated median of expression over a number of records as defined by a group by clause.

\section*{Example:}
```

Load Class, Median(Grade) as MG from abc.csv group by Class;

```

\section*{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.

\section*{Example:}
```

Load Key, sterr(X) as Z from abc.csv group by Key;

```

\section*{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 datapair will result in the entire data-pair to be disregarded.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 datapair will result in the entire data-pair to be disregarded.
An optional value \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point.
By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}

Load Key, linest_m(Y,X) as \(Z\) from abc.csv group by Key;

\section*{linest_b (y-expression, x-expression [, y0 [, x0 ]])}
returns the aggregated \(b\) value ( \(y\)-intercept) of a linear regression defined by the equation \(y=m x+b\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
```

Load Key, linest_b(Y,X) as Z from abc.csv group by Key;

```

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
```

Load Key, linest_r2(Y,X) as Z from abc.csv group by Key;

```

\section*{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 \(y=m x+b\) 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 valye y 0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
\[
\text { Load Key, linest_sem }(Y, X) \text { as } Z \text { from abc.csv group by Key; }
\]

\section*{linest_seb ( \(y\)-expression, x-expression \(\left[, y^{0}[, x 0]\right]\) )}
returns the aggregated standard error of the \(b\) value of a linear regression defined by the equation \(y=m x+b\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point.
By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}

Load Key, linest_seb \((Y, X)\) as \(Z\) from abc.csv group by Key;

\section*{linest_sey ( \(y\)-expression, \(x\)-expression \([, y 0[, x 0\) ] \(]\) )}
returns the aggregated standard error of the \(y\) estimate of a linear regression defined by the equation \(y=m x+b\)
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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
```

Load Key, linest_sey(Y,X) as Z from abc.csv group by Key;

```

\section*{linest_df ( \(y\)-expression, \(x\)-expression [, y0 [, x0 ]])}
returns the aggregated degrees of freedom of a linear regression defined by the equation \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
```

Load Key, linest_df(Y,X) as Z from abc.csv group by Key;

```

\section*{linest_f ( \(y\)-expression, \(x\)-expression [, y 0 [, x0 ]])}
returns the aggregated F statistic \(\left(\mathrm{r}^{2} /\left(1-\mathrm{r}^{2}\right)\right)\) of a linear regression defined by the equation \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y 0 may be stated forcing the regression line to pass through the y -axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}

Load Key, linest_f(Y,X) as \(Z\) from abc.csv group by Key;

\section*{linest_ssreg ( \(y\)-expression, \(x\)-expression [, y0 [, x0 ]])}
returns the aggregated regression sum of squares of a linear regression defined by the equation \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y 0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
```

Load Key, linest_ssreg(Y,X) as Z from abc.csv group by Key;

```

\section*{linest_ssresid (y-expression, x-expression \([, y 0\) [, x0 ]])}
returns the aggregated residual sum of squares of a linear regression defined by the equation \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y 0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.

\section*{Example:}
```

Load Key, linest_ssresid(Y,X) as Z from abc.csv group by Key;

```

\section*{Financial Aggregation Functions in Script}

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Example:}
```

Load Year, npv(0.05, Payments, PayDates) as PValue from abc.csv group
by Year;

```

\section*{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

\section*{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 295) function for argument descriptions.

\section*{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 295) function for argument descriptions.

\section*{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.

\section*{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 295) function for argument descriptions.

\section*{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 295) function for argument descriptions.

\section*{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 295) function for argument descriptions.

\section*{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 295) function for argument descriptions.

\section*{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 295) function for descriptions of the other arguments.

\section*{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 295) and TTest_conf (group, value [, sig \(=0.025\) [, eq_var \(=\) true]]) (page 296) functions for argument descriptions.

\section*{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 295) and TTest_conf (group, value [, sig \(=0.025\) [, eq_var \(=\) true]]) (page 296) functions for argument descriptions.

\section*{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.

\section*{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 297) function for argument descriptions.

\section*{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 297) function for argument descriptions.

\section*{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 297) function for argument descriptions.

\section*{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 297) function for argument descriptions.

\section*{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 297) function for descriptions of the other arguments.

\section*{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 297) and TTestlw_conf (weight, value [, sig \(=0.025\) ]) (page 300) functions for argument descriptions.

\section*{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 297) and TTestlw_conf (weight, value [, sig \(=0.025\) ]) (page 300) functions for argument descriptions.

\section*{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.

\section*{TTestl_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.

\section*{Example:}
```

Load Year, ttest1_t(Value) as X from abc.csv group by Year;

```

\section*{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 TTestl_t (value) (page 298) function for argument descriptions.

\section*{Example:}

Load Year, ttest1_df(Value) as X from abc.csv group by Year;

\section*{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 TTestl_t (value) (page 298) function for argument descriptions.

\section*{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 298) function for argument descriptions.

\section*{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 TTest \(1_{-} t\) (value) (page 298) function for argument descriptions.

\section*{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 TTestl_t (value) (page 298) function for descriptions of the other arguments.

\section*{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 TTestl_t (value) (page 298) and TTest _conf (value [, sig \(=0.025\) ]) (page 299) functions for argument descriptions.

\section*{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 TTestl_t (value) (page 298) and TTestl_conf (value [, sig \(=0.025\) ]) (page 299) functions for argument descriptions.

\section*{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.

\section*{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.

\section*{Example:}

Load Year, ttestlw_t(Weight, Value) as X from abc.csv group by Year;

\section*{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 TTestl \(w_{-} t\) (weight, value) (page 299) function for argument descriptions.

\section*{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 299) function for argument descriptions.

\section*{Example:}
```

Load Year, ttest1w_sig(Weight, Value) as X from abc.csv group by Year;

```

\section*{TTest1w_dif (weight, value)}
returns the aggregated student's t-test mean difference for a series of values iterated over a number of records as defined by a group by clause. See the TTestlw_t (weight, value) (page 299) function for argument descriptions.

\section*{Example:}
```

Load Year, ttest1w_dif(Weight, Value) as X from abc.csv group by Year;

```
```

TTest1w_sterr (weight, value)

```
returns the aggregated student's \(t\)-test standard error of the mean difference for a series of values iterated over a number of records as defined by a group by clause. See the TTestlw_t (weight, value) (page 299) function for argument descriptions.

\section*{Example:}

Load Year, ttest1w_sterr(Weight, Value) as X from abc.csv group by Year;
```

TTest1w_conf ( weight, value [, sig = 0.025 ])

```
returns the aggregated \(t\) value for a series of values iterated over a number of records as defined by a group by clause. The two-tailed level of significance can be specified in sig. If omitted sig will be set to 0.025 , resulting in a \(95 \%\) confidence interval. See the TTest \(1 w_{-} t\) (weight, value) (page 299) function for descriptions of the other arguments.

\section*{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 299) and TTestlw_conf (weight, value [, sig \(=0.025\) ]) (page 300) functions for argument descriptions.

\section*{Example:}
```

Load Year, ttest1w_lower(Weight, Value) as X from abc.csv group by
Year;

```
```

TTest1w_upper (weight, value [, sig = 0.025 ])

```
returns the aggregated value for the upper end of the confidence interval for a series of values iterated over a number of records as defined by a group by clause. See the TTestlw_t (weight, value) (page 299) and TTestlw_conf (weight, value [, sig = 0.025 ]) (page 300) functions for argument descriptions.

\section*{Example:}
```

Load Year, ttestlw_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.

\section*{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 301) function for argument descriptions.
returning null.

\section*{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 301) function for argument descriptions.
returning null.

\section*{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 301) function for argument descriptions.
returning null.

\section*{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 301) function for descriptions of the other arguments.
returning null.

\section*{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.

\section*{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 rec-
 argument descriptions.

\section*{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 302) function for argument descriptions.

\section*{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 302) function for argument descriptions.

\section*{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 302) function for descriptions of the other arguments.

\section*{Example:}
```

Load Year, ztestw_conf(Weight,Value-TestValue) as X from abc.csv group
by Year;

```

Statistical Test Functions in Script (page 295)

\subsection*{25.3 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:
\begin{tabular}{ll} 
Chart General Numeric Functions & page 805 \\
Chart Range Functions & page 809 \\
Exponential and Logarithmic Functions & page 312 \\
Trigonometric and Hyperbolic Functions & page 313 \\
Statistical Distribution Functions & page 314 \\
Financial Functions & page 315 \\
Mathematical Constants and Parameter-Free Functions & page 318 \\
String Functions & page 320 \\
Chart Inter Record Functions & page 815 \\
Document Functions & page 337 \\
Chart Ranking Functions & page 821 \\
Conditional Functions & page 328 \\
Logical Functions & page 329 \\
NULL Functions & page 330 \\
System Functions & page 330 \\
Date and Time Functions & page 338 \\
Interpretation Functions & page 825 \\
Formatting Functions & page 356 \\
Color Functions & page 361
\end{tabular}

\section*{General Numeric Functions}

In the functions below, the parameters are expressions where \(x\) should be interpreted as a real valued number.

\section*{\(\operatorname{div}(x 1, x 2)\)}

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.

\section*{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. \(x 2\) must be greater than 0 . The result is the non-negative remainder of an integer division.

\section*{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 NUL

```

\section*{fmod (x1 , \(x 2\) )}

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.

\section*{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.

\section*{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

```

\section*{floor(x [ , base [ , offset ]])}

Rounding of \(x\) downwards to the nearest multiple of base with an offset of offset. The result is a number.

\section*{Examples:}
```

floor( 2.4 ) returns 2
floor( 2.6 ) returns2
floor( 3.88 , 0.1 ) returns 3.8
floor( 3.88 , 5 ) returns 0

```
```

floor( 1.1 , 1 , 0.5 ) returns 0.5

```

\section*{frac (x)}

Returns the fraction part of \(x\). The fraction is defined in such a way that \(\operatorname{frac}(x)+\boldsymbol{f l o g}(x)=x\).

\section*{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.

\section*{Examples:}
```

round( 2.4 ) returns2
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

```

\section*{fabs (x)}

The absolute value of \(x\). The result is a positive number.

\section*{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 nonnumeric values as 0 .

\section*{Examples:}
```

numsum( 1,2,4 ) returns 7
numsum( 1,'xyz' ) returns 1
numsum( null( ) ) returns 0

```

\section*{Note!}

The numsum function is now obsolete and is superseded by the rangesum (expr \(1[, \operatorname{expr} 2, \ldots \operatorname{expr} N]\) ) (page 308) function.
```

numcount(expr1 [ , expr2, ... exprN ])

```

Returns the number of numeric values found in 1 to N arguments

\section*{Examples:}
numcount ( \(1,2,4\), ) returns 3
```

numcount( 2,xyz ) returns 1
numcount( null( ) ) returns 0

```

\section*{Note!}

The numcount function is now obsolete and is superseded by the rangecount(expr1 [, expr2, ... exprN ]) (page 308) function.
```

numavg(expr1 [ , expr2, ... exprN ])

```

Returns the numeric average of 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
```

numavg( 1,2,4 ) returns 2.33333333
numavg( 1,'xyz' ) returns 1
numavg( null( ) 'abc' ) returns NULL

```

\section*{Note!}

The numavg function is now obsolete and is superseded by the rangeavg(expr1 [, expr2, ... exprN ]) (page 308) function.
```

nummin(expr1 [ , expr2, ... exprN ])

```

Returns the lowest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
```

nummin( 1,2,4 ) returns 1
nummin( 1,'xyz' ) returns 1
nummin( null( ) 'abc' ) returns NULL

```

\section*{Note!}

The nummin function is now obsolete and is superseded by the rangemin(exprl [, expr2, \(\ldots\) exprN ]) (page 308) function.
```

nummax(expr1 [ , expr2, ... exprN ])

```

Returns the highest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
```

nummax( 1,2,4 ) returns 4
nummax( 1,'xyz' ) returns 1
nummax( null( ) 'abc' ) returns NULL

```

\section*{Note!}

The nummax function is now obsolete and is superseded by the rangemax (exprl [, expr2, \(\ldots\) exprN ]) (page 309) function.

\section*{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.

\section*{Examples:}
fact( 1 ) returns 1
```

fact( 5 ) returns 120(1*2*3*4*5=120)

```
fact ( -5 ) returns NULL

\section*{combin(n1, n2)}

Returns the number of combinations of \(n 2\) items that can be picked from a group of \(n 1\) items. The order in which the items are selected is insignificant. Non-integer items will be truncated.

\section*{Example:}

How many combinations of 7 numbers can be picked from a total of 35 Lotto numbers?
combin( 35,7 ) returns 6724520
```

permut(n1, n2)

```

Returns the number of permutations of \(n 2\) items that can be selected from a group of \(n 1\) items. The order in which the items are selected is significant. Non-integer arguments will be truncated.

\section*{Example:}

In how many ways could the gold, silver and bronze medals be distributed after a 100 m final with 8 participants?
permut ( 8,3 ) returns 336

\section*{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.

\section*{Examples:}
even ( 3 ) returns false
even ( 2 * 10 ) returns true
even ( 3.14 ) returns NULL

\section*{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.

\section*{Examples:}
```

odd( 3 ) returns true
odd( 2 * 10 ) returns false
odd( 3.14 ) returns NULL

```

\section*{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.

\section*{Examples:}
```

sign( 66 ) returns 1
sign( 0 ) returns 0
sign( - 234 ) returns-1

```

\section*{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.

\section*{Examples:}
```

bitcount ( 3 ) returns2
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.

\section*{Script Range Functions}

Range functions will replace the following General Numeric Functions (page 303): 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 nonnumeric values as 0 .

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Examples:}
rangecorrel \((2,3,6,8,9,4)\) returns 0,269

\section*{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 815) 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.

\section*{Examples:}
```

rangeirr(-70000,12000,15000,18000,21000,26000) returns 0,0866
rangeirr(above(sum(value), 0, 10))
rangeirr(above(total value, 0, rowno(total)))

```

\section*{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 815) with a third optional parameter. Text values, null values and missing values are disregarded.

\section*{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 311) 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 815) 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.

\section*{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)))

```

\section*{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 311) 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 815) 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.

\section*{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.

\section*{Exponential and Logarithmic Functions}

In the functions below, the parameters are expressions where \(x\) should be interpreted as a real valued number.

\section*{\(\exp (x)\)}

Exponential function, with the base of the natural \(\operatorname{logarithm} e\) as base. The result is a positive number.

\section*{\(\log (x)\)}

The natural logarithm of \(x\). The function is only defined if \(x>0\). The result is a number.

\section*{\(\log 10(x)\)}

The 10-logarithm (base 10) of \(x\). The function is only defined if \(x>0\). The result is a number.

\section*{sqrt(x )}

Square root of \(x\). The function is only defined if \(x>=0\). The result is a positive number.

\section*{sqr ( \(x\) )}

Square of \(x\). The result is a number.

\section*{pow ( \(x, y\) )}

Returns \(x\) to the power of \(y\). The result is a number.

\section*{Back to Other Functions}

\section*{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.

\section*{\(\cos (x)\)}

Cosine of \(x\). The result is a number between -1 and 1 .

\section*{\(\operatorname{acos}(x)\)}

Arcus cosine of \(x\). The function is only defined if \(-1 \leq \leq_{1}\). The result is a number between 0 and \(\pi\).

\section*{\(\sin (x)\)}

Sine of \(x\). The result is a number between -1 and 1 .

\section*{\(\operatorname{asin}(x)\)}

Arcus sine of \(x\). The function is only defined if \(-1 \leq \leq_{x}\). The result is a number between \(-\pi / 2\) and \(\pi / 2\).
\(\tan (x)\)
Tangent of \(x\). The result is a number.

\section*{\(\operatorname{atan}(x)\)}

Arcus tangent of \(x\). The result is a number between \(-\pi / 2\) and \(\pi / 2\).
\(\operatorname{atan} 2(y, x)\)
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 \(+\pi\).

\section*{\(\cosh (x)\)}

Cosine hyberbolicus of \(x\). The result is a positive number.
\(\sinh (x)\)
Sine hyberbolicus of \(x\). The result is a number.

\section*{\(\tanh (x)\)}

Tangent hyberbolicus of \(x\). The result is a number.

Back to Other Functions.

\section*{Statistical Distribution Functions}

The statistical distribution functions described below are all implemented in QlikView using the Cephes library. For references and details on algorithms used, accuracy etc, please see http://www.netlib.org/cephes/. The Cephes function library is used by permission.
```

chidist (value, degrees freedom)

```
returns the one-tailed probability of the chi2 distribution. The Chi 2 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 \(=\operatorname{chidist}(\) value, df\()\), then \(\operatorname{chiinv}(\) prob, df\()=\) value .

\section*{Example:}
```

chidist( 8, 15 ) returns 0.9237827

```

\section*{chiinv (prob, degrees_freedom)}
returns the inverse of the one-tailed probability of the chi 2 distribution. Prob is a probability associated with the chi 2 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 314) function in the following way:
If prob \(=\operatorname{chidist}(\) value, df\()\), then \(\operatorname{chiinv}(\) prob, df\()=\) value .

\section*{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 314) function in the following way:
If prob \(=\) normdist \((\) value \(, m, s d)\), then norminv \((\operatorname{prob}, m, s d)=\) value .

\section*{Example:}
chdist ( \(0.5,0,1\) ) returns 0.6914625
norminv (prob, mean, standard_dev)
returns the inverse of the normal cumulative distribution for the specified mean and standard deviation. Prob is a probability associated with the normal distribution. It must be a number between 0 and 1 . Mean is a value stating the arithmetic mean for the distribution. Standard_dev is a positive value stating the standard deviation of the distribution. All arguments must be numeric, else null will be returned. This function is related to the normdist (value, mean, standard_dev) (page 314) function in the following way:
If prob \(=\) normdist(value, \(m, s d)\), then norminv(prob, \(m, s d)=\) value .

\section*{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 tinv (prob, degrees_freedom) (page 315) function in the following way:
If prob \(=\operatorname{tdist}(\) value, \(\mathrm{df}, 2)\), then \(\operatorname{tinv}(\) prob, df\()=\) value .

\section*{Example:}
chdist( 1, 30, 2 ) returns 0.3253086
```

tinv (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 314) function in the following way:
If prob \(=\operatorname{tdist}(\) value, \(\mathrm{df}, 2)\), then \(\operatorname{tinv}(\) prob, df\()=\) value .

\section*{Example:}
tinv(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 freedom 2 is a positive integer stating the number of denominator degrees of freedom. All arguments must be numeric, else null will be returned. This function is related to the finv (prob, degrees_freedom 1, degrees freedom2) (page 315) function in the following way:
If prob \(=\) fdist(value, \(\mathrm{df} 1, \mathrm{df} 2)\), then \(\mathrm{finv}(\) prob, \(\mathrm{df} 1, \mathrm{df} 2)=\) value .

\section*{Example:}
fdist ( 15, 8, 6 ) returns 0.0019369

\section*{finv (prob, degrees_freedom1, degrees_freedom2)}
returns the inverse of the F-distribution. Prob is a probability associated with the F-distribution and must be a number between 0 and 1. Degrees_freedom is an integer stating the number of degrees of freedom. All arguments must be numeric, else null will be returned. This function is related to the fdist (value, degrees_ freedom 1, degrees_freedom2) (page 315) function in the following way:
If prob \(=\) fdist \((\) value, \(d f 1, d f 2)\), then \(\operatorname{finv}(\) prob, \(d f 1, d f 2)=\) value .

\section*{Example:}
```

finv( 0.0019369, 8, 5 ) returns 15.0000197

```

\section*{Back to Other Functions.}

\section*{Financial Functions}

\footnotetext{
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.
\(p m t\) is the payment made each period. It cannot change over the life of the annuity. If \(p m t\) is omitted, the \(p v\) argument must be included.
\(p v\) is the present value, or lump-sum amount, that a series of future payments is worth right now. If \(p v\) is omitted, it is assumed to be 0 (zero) and the pmt argument must be included.
\(f v\) is the future value, or cash balance, you want to attain after the last payment is made. If \(f v\) 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 .

\section*{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 357).

\section*{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?
\[
\mathrm{fv}(0.005,36,-20) \text { returns } \$ 786.72
\]
```

nper(rate, pmt, pv [ ,fv [ , type ] ])

```

Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.

\section*{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 357).
To find the total amount paid over the duration of the loan, multiply the returned pmt value by nper.

\section*{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 357). 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.

\section*{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.

\section*{Example:}

What is the interest rate of a five-year \(\$ 10,000\) annuity loan with monthly payments of \(\$ 300\) ?
rate \((\mathbf{6 0}, \mathbf{- 3 0 0 , 1 0 0 0 0})\) returns \(2.18 \%\)

Back to Other Functions.

\section*{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 318) 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.

\section*{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 \(4 \%\) )

\section*{The Black and Schole Formula}

\section*{For \(x \geq 0\)}
\[
\begin{aligned}
& \int_{-\infty}^{x} e^{-u^{2} / 2} d u \approx \sqrt{2 \pi}-\left(b_{1} k+b_{2} k^{2}+b_{3} k^{3}\right) e^{-x^{2} / 2} \\
& k=\frac{1}{1+0.33267 x} \quad b=\left(\begin{array}{c}
0.4361836 \\
-0.1201676 \\
0.937298
\end{array}\right)
\end{aligned}
\]

\section*{For \(x<0\)}
\(f(x)=1-f(-x)\).

\section*{Back to Other Functions.}

\section*{Mathematical Constants and Parameter-Free Functions}

These do not have any parameters. The parentheses are however still required.

\section*{e()}

Base of the natural logarithms, \(e\). The function returns 2.71828...

\section*{pi()}
\(\pi\) The function returns \(3.14159 \ldots\)

\section*{rand( )}

Returns a random number between 0 and 1 .

\section*{true ( )}

Returns a dual value with text value 'true' and numeric value -1 , which can be used as logical true in expressions.

\section*{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.

\section*{Counter Functions}

These functions do not have any parameters. The parentheses are however still required.

\section*{RecNo ( )}

Returns an integer for the number of the currently read row of the internal table. The first record is number 1.

\section*{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 Concatenation to another. The first row is number 1.

\section*{Examples:}

Raw data tables:

\section*{Tab1.csv}

A B

1 aa

2 cc

3 ee

\section*{Tab2.csv}

A B
\(5 \quad x x\)

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:
\begin{tabular}{cccc} 
QVTab & & & \\
A & B & \(\operatorname{RecNo}()\) & RowNo( ) \\
1 & aa & 1 & 1 \\
3 & ee & 3 & 2 \\
4 & yy & 2 & 3 \\
6 & zz & 3 & 4
\end{tabular}

\section*{IterNo( )}

This function is only meaningful if used together with a while clause, see Load (page 246). 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.

\section*{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.

\section*{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 the 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.

\section*{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.

\section*{Example:}

Autonumberhash256 ( Region, Year, Month )

\section*{fieldvaluecount(fieldname)}

Returns the number of distinct values in a field. Fieldname must be given as a string (e.g. a quoted literal).

\section*{Example:}
```

let x = fieldvaluecount('Alfa');

```

\section*{Back to Other Functions.}

\section*{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.

\section*{Example:}
ord ('A') returns the number 65 .
```

chr( n )

```

The ASCII character corresponding to number \(n\). The result is a string.

\section*{Example:}
chr (65) returns the string ' \(A\) '.

\section*{len ( \(s\) )}

Length of string \(s\). The result is an integer.

\section*{Example:}
len (Name) where Name \(=\) 'Peter' returns 5.

\section*{left( \(s, n\) )}

Substring of the string \(s\). The result is a string consisting of the first \(n\) characters of \(s\).

\section*{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.

\section*{right ( \(s, n\) )}

Substring of the string \(s\). The result is a string consisting of the last \(n\) characters of \(s\).

\section*{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 \(n l\) with the length of \(n 2\) characters. If \(n 2\) is omitted, the function returns the rightmost part of the string starting at character \(n 1\). The positions in the string are numbered from 1 and up.

\section*{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.

\section*{index (s1, s2[, n])}

Position of a substring. This function gives the starting position of the \(n:\) th occurrence of substring \(s 2\) in string \(s l\). If \(n\) is omitted, the first occurrence is assumed. If \(n\) is negative, the search is made starting from the end of string \(s l\). The result is an integer. The positions in the string are numbered from \(l\) and up.

\section*{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

\section*{upper(textexpression)}

Forces upper case for all data in the expression.

\section*{Example:}
```

upper( 'abcD' ) returns 'ABCD'.

```

\section*{lower ( textexpression )}

Forces lower case for all data in the expression.

\section*{Example:}
lower ( 'abcD' ) returns 'abcd'.
```

repeat( }s,n

```

Forms a string consisting of the string \(s\) repeated \(n\) times.

\section*{Example:}
```

repeat( ' * ', rating ) when rating=4 returns '****'.

```

\section*{ltrim (s )}

Returns the string \(s\) trimmed of any leading spaces.

\section*{Example:}
```

ltrim ( ' abc' ) returns 'abc'.
ltrim ( 'abc ' ) returns'abc'

```

\section*{rtrim (s )}

Returns the string s trimmed of any trailing spaces.

\section*{Example:}
```

rtrim ( ' abc' ) returns'abc'
rtrim ( 'abc ' ) returns 'abc'

```

\section*{trim(s )}

Returns the string \(s\) trimmed of any leading and trailing spaces.

\section*{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.

\section*{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 ';'

```

\section*{KeepChar (s1 , s2)}

Returns the string \(s l\) less all characters not contained in string \(s 2\).

\section*{Example:}
```

keepchar ( 'a1b2c3','123' ) returns'123'

```

\section*{PurgeChar (s1, s2)}

Returns the string \(s l\) less all characters contained in string \(s 2\).

\section*{Example:}
purgechar ( 'a1b2c3','123' ) returns 'abc'

\section*{capitalize (s )}

Returns the string \(s\) with all words capitalized.

\section*{Example:}
```

capitalize ('my little pony') returns 'My Little Pony'
capitalize ( 'AA bb cC Dd') returns 'Aa Bb Cc Dd'

```

\section*{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.

\section*{Note!}

This string function can not be used in chart expressions.

\section*{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 \(s\).

\section*{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.

\section*{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.

\section*{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', 'a%\&') returns'0'

```

\section*{hash128(expression \{, expression\})}

Returns a 128 -bit hash of the combined input expression values. The result is a string.

\section*{Examples:}
```

hash128 ( 'abc', 'xyz', '123' )
hash128 ( Region, Year, Month )

```

\section*{hash160(expression \{, expression\})}

Returns a 160 -bit hash of the combined input expression values. The result is a string.

\section*{Example:}
hash160 ( Region, Year, Month )

\section*{hash256(expression \{, expression\} )}

Returns a 256-bit hash of the combined input expression values. The result is a string.

\section*{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

\section*{Example:}
substringcount ( 'abcdefgcdxyz', 'cd' ) returns 2

\section*{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.

\section*{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 253)). Its name must be enclosed by single, straight Quotation Marks (page 281).
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.

\section*{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 253)). The name must be enclosed by single straight quotation marks. Expr is the expression whose result should be mapped by substrings.

\section*{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'
MapSu.bstring ('map1', 'baaar') returns 'bXYZar'
MapSubstring ('map1', 'xaa1') returns 'bXYZ<one>'

```

\section*{Back to Other Functions}

\section*{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.

\section*{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.

\section*{previous (expression )}

Returns the value of expression using data from the previous input record that was not discarded due to a where clause. In the first record of an 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.

\section*{Examples:}
```

Load *, Sales / previous(Sales) as Increase from ...;
Load A, previous(previous( A )) as B from ...;

```

\section*{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 273), 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 tablename must be included.

\section*{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.

\section*{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.

\section*{Example:}
```

FieldValue( 'HelpText', 5 )

```

\section*{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.

\section*{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.

\section*{Example:}
```

lookup('Price', 'ProductID', InvoicedProd, 'pricelist')

```

Back to Other Functions.

\section*{Conditional Functions}

\section*{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.

\section*{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.

\section*{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, exprl[ , expr2,...exprN])

```

Returns the \(n\) :th expression in the list. \(n\) is an integer between 1 and \(N\).

\section*{Example:}
```

    pick( N'A''B'4, , , )
    ```
        returns ' \(B^{\prime}\) if \(N=2\)
        returns 4 if \(N=3\)
```

match( str, expr1 [ , expr2,...exprN ] )

```

The match function performs a case sensitive comparison.

\section*{Example:}
\[
\begin{gathered}
\text { match (M, 'Jan', 'Feb', 'Mar') } \\
\text { returns } 2 \text { if } M=F e b \\
\text { returns } 0 \text { if } M=A p r \text { or } j a n
\end{gathered}
\]
```

mixmatch( str, expr1 [ , expr2,...exprN ] )

```

The mixmatch function performs a case insensitive comparison.

\section*{Example:}
```

mixmatch( M, 'Jan','Feb','Mar')
returns l if M=jan

```
```

wildmatch( str, exprl [ , expr2,..exprN] )

```

The wildmatch function performs a case insensitive comparison and permits the use of wildcard characters ( * and ?) in the comparison strings.

\section*{Example:}
```

wildmatch( M, 'ja*','fe?','mar')
returns l if M= January
returns2 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 \(\mathrm{a}<=\mathrm{x}<\mathrm{b}\), 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.

\section*{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.

\section*{Logical Functions}

\section*{IsNum ( expr)}

Returns -1 (true) if the expression can be interpreted as a number, otherwise 0 (false).

\section*{IsText (expr)}

Returns -1 (true) if the expression has a text representation, otherwise 0 (false).

\section*{IsPartialReload( )}

Returns - 1 (true) if the current reload is partial, otherwise 0 (false).

\section*{Back to Other Functions.}

\section*{NULL Functions}

\section*{Null( )}

Returns a real NULL value.

\section*{IsNull (expr )}

Returns -1 (true) if expr returns NULL, otherwise 0 (false).

Back to Other Functions.

\section*{System Functions}

\section*{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.

\section*{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.

\section*{OSuser ( )}

Returns a string containing the name of the current user as returned by the operating system.

\section*{QVuser ( )}

Returns a string containing the name of the current QlikView user as entered in a section access.

\section*{ComputerName ( )}

Returns a string containing the name of the computer as returned by the operating system.

\section*{ReloadTime( )}

Returns a timestamp for when the script last finished reexecuting.

\section*{GetActiveSheetID()}

Returns a string containing the Id of the active sheet.

\section*{GetCurrentField (groupname)}
returns the name of the field currently active in the group named groupname.

\section*{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.

\section*{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.

\section*{Examples:}
```

getfieldselections(Country)
getfieldselections (Country, ';' )
getfieldselections (Country, ',' , 10 )

```
```

GetSelectedCount(fieldname [, includeexcluded=false])

```
```

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.

\section*{Examples:}
```

getselectedcount( Customers )
getselectedcount( Customers, true )

```

\section*{GetPossibleCount(fieldname)}
returns the number of possible values in the field named fieldname.

\section*{Example:}
```

    getpossiblecount( Customers )
    ```

\section*{GetExcludedCount(fieldname)}
returns the number of excluded values in the field named fieldname.

\section*{Example:}
getexcludedcount( Customers )

\section*{GetAlternativeCount(fieldname)}
returns the number of alternative values in the field named fieldname. Read about alternative values in Color Scheme (page 107).

\section*{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

\section*{Examples:}
```

getnotselectedcount( Country )
getnotselectedcount( Country, true )

```

\section*{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.

\section*{Example:}
```

getregistrystring('HKEY_LOCAL_MACHINE\SOF-
TWARE\QlikTech\QlikViewS̄erver<br>Settings 7','EnableSessionLog')

```

\section*{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_def-
button]]]] )

```

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,
6for 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.

\section*{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.

\section*{Example:}
```

Load
Input('Enter value', 'Input box') as v,
Recno () as r
autogenerate 3;

```

\section*{DocumentName ( )}

Returns a string containing the name of the current QlikView document, without path but with extension.

\section*{DocumentPath ( )}

Returns a string containing the full path to the current QlikView document.

\section*{DocumentTitle( )}

Returns a string containing the title of the current QlikView document.

\section*{GetObjectField ([index])}
returns the name of the dimension. Index is an optional integer denoting which of the used dimensions that should be returned.

\section*{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.

\section*{Example:}
```

Dynamic Text
='Region - ' \& if(StateName() = '\$', 'Default', StateName())

```

\section*{Example:}

Dynamic Colors
```

if(StateName() = 'Group 1', rgb (152, 171, 206),
if(StateName() = 'Group 2', rgb(187, 200, 179),
rgb(210, 210, 210)
)
)

```

\section*{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.
Extended property is defined for extension object in its definition file.

\section*{Example:}
```

GetExtendedProperty ('Greeting')

```

\section*{Back to Other Functions.}

\section*{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:

\section*{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 \(+\mathrm{Q}, \mathrm{J}, \mathrm{P}, \mathrm{G}\) for jpg files (keep the CTRL key pressed while typing the QJPG combination), CTRL \(+\mathrm{Q}, \mathrm{M}, \mathrm{P}, 3\) for mp3 files and CTRL \(+\mathrm{Q}, \mathrm{W}, \mathrm{M}, \mathrm{A}\) for wma files.

\section*{Example:}

Attribute('File', 'Title') as X, returns the mp3 tag 'title' in field \(X\) in each record.

\section*{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.

\section*{filebasename ()}

Returns a string containing the name of the table file currently being read, without path or extension.

\section*{Example:}
```

Load *, filebasename( ) as X from
C:\UserFiles\abc.txt

```

Will return 'abc' in field X in each record read.

\section*{filedir()}

Returns a string containing the path to the directory of the table file currently being read.

\section*{Example:}
```

Load *, filedir( ) as X from
C:\UserFiles\abc.txt
Will return ' $\mathrm{C}: \backslash \mathrm{UserFiles'} \mathrm{in} \mathrm{field} \mathrm{X}$ in each record read.

```

\section*{fileextension()}

Returns a string containing the extension of the table file currently being read.

\section*{Example:}
```

Load *, fileextension( ) as X from

```
```

C:\UserFiles\abc.txt

```

Will return 'txt' in field X in each record read.

\section*{filename()}

Returns a string containing the name of the table file currently being read, without path but including the extension.

\section*{Example:}
```

Load *, filename( ) as X from
C:\UserFiles\abc.txt
Will return 'abc.txt' in field X in each record read.

```

\section*{filepath()}

Returns a string containing the full path to the table file currently being read.

\section*{Example:}
```

Load *, filepath( ) as X from

```

C: \UserFiles\abc.txt
Will return ' \(\mathrm{C}: \backslash\) UserFiles \(\backslash a b c . t x t\) ' in field X in each record read.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Examples:}
```

GetFolderPath('MyMusic')
GetFolderPath('MyPictures')
GetFolderPath('MyVideos')
GetFolderPath('MyReceivedFiles')
GetFolderPath('MyShapes')

```
```

GetFolderPath('ProgramFiles')
GetFolderPath('Windows')

```

\section*{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.

\section*{Examples:}
```

QvdCreateTime('MyFile.qvd')
QvdCreateTime('C:\MyDir\MyFile.qvd')

```

\section*{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.

\section*{Examples:}
```

QvdNoOfRecords ('MyFile.qvd')
QvdNoOfRecords ('C:\MyDir\MyFile.qvd')

```

\section*{QvdNoOfFields(filename)}

Returns the number of fields in a QVD file.
The filename is the name of a QVD file, if necessary including path.

\section*{Examples:}
```

QvdNoOfFields ('MyFile.qvd')
QvdNoOfFields ('C:\MyDir\MyFile.qvd')

```

\section*{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.

\section*{Examples:}
```

QvdFieldName ('MyFile.qvd', 3)
QvdFieldName ('C:\MyDir\MyFile.qvd', 5)

```

\section*{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.

\section*{Examples:}

QvdTableName ('MyFile.qvd')
QvdTableName ('C: \MyDir\MyFile.qvd')

Back to Other Functions.

\section*{Table Functions}

The table functions return information about the data table which is currently being read. If no table name is specified and the function is used within a load statement, the current table is assumed.

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.

\section*{Examples:}
```

LET a = FieldName(4,'tab1');
T1:
Load a, b, c, d from abc.csv
T2:
Load FieldName (2, 'T1') Autogenerate 1;

```

\section*{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.

\section*{Examples:}
```

LET a = FieldNumber('Customer','tab1');
T1:
Load a, b, c, d from abc.csv
T2:
Load FieldNumber ('b', 'T1') Autogenerate 1;

```

\section*{NoOfFields([ 'TableName ' ])}

Returns the number of fields in a previously loaded table. If the function is used within a load statement, it must not reference the table currently being loaded.

\section*{Examples:}
```

LET a = NoOfFields('tab1');
Load *, NoOfFields( ) from abc.csv;

```

\section*{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.

\section*{Examples:}
```

LET a = NoOfRows('tab1');
Load * from abc.csv where NoOfRows( ) < 30;

```

\section*{NoOfTables()}

Returns the number of tables previously loaded.

\section*{TableName ([ 'TableNumber' ])}

Returns the name of the table with the specified number.

\section*{TableNumber([ 'TableName' ])}

Returns the number of the specified table.

Back to Other Functions.

\section*{Document Functions}

These functions can be used in charts and script.

\section*{ReportComment(report_number)}

Returns the comment of the report with the specified number within the active document.

\section*{ReportName (report_number)}

Returns the name of the report with the specified number within the active document.

\section*{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.

\section*{NoOfReports()}

Returns the number of reports in the active document.

\section*{Back to Other Functions.}

\section*{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.

\section*{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.

\section*{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 828) interpretation function.

\section*{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.

\section*{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 338).

\section*{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.

\section*{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 338).

\section*{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.

\section*{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 827) interpretation function.

\section*{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.

\section*{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 339).

\section*{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.

\section*{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 339).

\section*{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.

\section*{Example:}
\[
\text { year ( '1971-10-30' ) returns } 1971 .
\]

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 339).

\section*{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.

\section*{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 339).

\section*{weekday (date)}

Week day. Returns an integer between 0-6.

\section*{Example:}
\[
\text { weekday ( '1971-10-30' ) returns } 5 .
\]

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 339).

\section*{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 script run
1 Time at function call
2 Time when the document was opened
Default timer_mode is 1 . The timer_mode \(=1\) should be used with caution, since it polls the operating system every second and hence could slow down the system.

\section*{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.

\section*{LocalTime ([timezone [, ignoreDST ]])}

Returns a timestamp of the current time from the system clock for a specified time zone. The timezone is specified as a string containing any of the geographical places listed under Time Zone in the Windows Control Panel for Date and Time or as a string in the form 'GMT+hh:mm'. If no timezone is specified the local time will be returned. If ignoreDST is -1 (true) daylight savings time will be ignored.

\section*{Examples:}
```

localtime ('Paris')
localtime ('GMT+01:00')
localtime ('Paris',-1)
localtime()

```

\section*{MakeDate (YYYY [ , MM [ , DD ] ] )}

Returns a date calculated from the year \(Y Y Y Y\), the month \(M M\) and the day \(D D\).
If no month is stated, 1 (January) is assumed.
If no day is stated, 1 (the 1 :st) is assumed.

\section*{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

\section*{MakeWeekDate (YYYY [ , WW [ , D ] ] )}

Returns a date calculated from the year \(Y Y Y Y\), the week \(W W\) and the day-of-week \(D\).
If no day-of-week is stated, 0 (Monday) is assumed.

\section*{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 \(h h\), the minute \(m m\) the second \(s s\) with a fraction \(f f f\) 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.

\section*{Examples:}
```

maketime( 22 ) returns 22-00-00
maketime( 22, 17 ) returns 22-17-00
maketime( 22, 17, 52 ) returns 22-17-52

```

\section*{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 \(=1\) ).

\section*{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'

```

\section*{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 calender 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.

\section*{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

```

\section*{TimeZone ( )}

Returns the name of the current time zone, as defined in Windows.

\section*{Example:}
```

    timezone( )
    ```
GMT ( )

Returns the current Greenwich Mean Time, as derived from the system clock and Windows time settings.

\section*{Example:}
gmt ( )

\section*{UTC ( )}

Returns the current Coordinated Universal Time.

\section*{Example:}
utc ( )

\section*{DaylightSaving( )}

Returns the current adjustment for daylight saving time, as defined in Windows.

\section*{Example:}
```

daylightsaving( )

```

\section*{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.

\section*{Examples:}
```

setdateyear ('2005-10-29', 2006) returns '2006-10-29'
setdateyear ('2005-10-29 04:26', 2006) returns '2006-10-29 04:26'

```

\section*{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.

\section*{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'

\section*{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.

\section*{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

```

\section*{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.

\section*{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

\section*{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.

\section*{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.

\section*{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

```

\section*{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.

\section*{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.

\section*{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 1 st ) 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.

\section*{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 1 st) 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.

\section*{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

\section*{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.

\section*{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

\section*{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.

\section*{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

```

\section*{InLunarWeek (date, basedate , shift [, weekstart])}
returns true if date lies inside the lunar week (consecutive 7 day periods starting on January 1 st 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.

\section*{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

```

\section*{InLunarWeekToDate (date, basedate, shift [, weekstart])}
returns true if date lies inside the part of lunar week (consecutive 7 day periods starting on January 1 st 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.

\section*{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 3 am .

\section*{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.

\section*{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

\section*{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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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.

\section*{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'

\section*{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 1 st ) 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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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'

\section*{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.

\section*{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.

\section*{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 1 st 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.

\section*{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'

\section*{LunarweekEnd(date [, shift \(=0 \quad\) [, weekoffset \(=0]])\)}

Returns a value corresponding to a timestamp with the last millisecond of the lunar week (consecutive 7 day periods starting on January 1 st 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.

\section*{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'

\section*{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.

\section*{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 3 am .

\section*{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'

\section*{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.

\section*{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'

\section*{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 3 am .

\section*{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

\section*{age(timestamp, date_of_birth)}

Returns the age at the time of timestamp (in completed years) of somebody born on date_of_birth.

\section*{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.

\section*{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

\section*{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.

\section*{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'

\section*{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.

\section*{Examples:}
lastworkdate ('2007-02-19', 9) returns '2007-03-01'
lastworkdate ('2006-12-18', 8, '2006-12-25', '2006-12-26') returns '2006-1229'

\section*{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().

\section*{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.

\section*{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 \(=3\).

\section*{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.

\section*{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 \(=3\).

\section*{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

\section*{Back to Other Functions.}

\section*{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.

\section*{ThousandSep}

The thousands separator defined replaces the digit grouping symbol of the operating system (Regional Settings).

\section*{Example:}

Set ThousandSep=','; (for example, seven billion must be specified as: 7,000,000,000)

\section*{DecimalSep}

The decimal separator defined replaces the decimal symbol of the operating system (Regional Settings).

\section*{Example:}

Set DecimalSep='.';

\section*{MoneyThousandSep}

The thousands separator defined replaces the digit grouping symbol for currency of the operating system (Regional Settings).

\section*{Example:}
```

Set MoneyThousandSep=',';

```

\section*{MoneyDecimalSep}

The decimal separator defined replaces the decimal symbol for currency of the operating system (Regional Settings).

\section*{Example:}

Set MoneyDecimalSep='.';

\section*{MoneyFormat}

The symbol defined replaces the currency symbol of the operating system (Regional Settings).

\section*{Example:}
```

Set MoneyFormat='\$ \#,\#\#0.00; (\$ \#,\#\#0.00)';

```

\section*{TimeFormat}

The format defined replaces the time format of the operating system (Regional Settings).

\section*{Example:}
```

Set TimeFormat='hh:mm:ss';

```

\section*{DateFormat}

The format defined replaces the date format of the operating system (Regional Settings).

\section*{Example:}

Set DateFormat='M/D/YY';

\section*{TimestampFormat}

The format defined replaces the date and time formats of the operating system (Regional Settings).

\section*{Example:}

Set TimestampFormat='M/D/YY hh:mm:ss[.fff]';

\section*{MonthNames}

The format defined replaces the month names convention of the operating system (Regional Settings).

\section*{Example:}

Set MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';

\section*{LongMonthNames}

The format defined replaces the long month names convention of the operating system (Regional Settings).

\section*{Example:}

Set LongMonthNames='January;February;March;April;May;June - -

\section*{DayNames}

The format defined replaces the weekday names convention of the operating system (Regional Settings).

\section*{Example:}
```

Set DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

```

\section*{LongDayNames}

The format defined replaces the long weekday names convention of the operating system (Regional Settings).

\section*{Example:}
```

Set Long-
DayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';

```

Back to Other Functions.

\section*{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 434).

\section*{Note!}

For reasons of clarity all number representations are given with decimal point as decimal separator.

\section*{Back to Other Functions.}

\section*{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 set in the operating system is used.

\section*{Example:}

The examples below assume the two following operating system settings:
\begin{tabular}{lcc} 
& Default setting 1 & Default setting 2 \\
Number format & \(\# \# \# 0, \#\) & \(\#, \# \# 0 . \#\) \\
num( A, \(\mathbf{' 0 . 0}^{\prime}\) ) where \(A=35648.375\) returns: \\
Setting 1 & Setting 2 \\
String & 35648375 & 35648.375 \\
Number & 35648375 & 35648.375
\end{tabular}

\footnotetext{
num( A, '\#,\#\#0.\#\#', '.' , ',' ) where A=35648 returns:
}

Setting 1
35,648.00
35648

Setting 2
35,648.00
35648
num( \(\mathbf{p i (}),{ }^{\prime} \mathbf{0 , 0 0} \mathbf{'}^{\prime}\) ) returns:
Setting 1
Setting 2
String
3,14
3.141592653
3.141592653

Back to Other Functions.

\section*{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.

\section*{Examples:}

The examples below assume the two following operating system settings:
\begin{tabular}{ccc} 
& Default setting 1 & Default setting 2 \\
Money format & \(\mathrm{kr} \# \# \# 0,00\) & \(\$ \#, \# \# 0.00\)
\end{tabular}
money (A ) where A=35648 returns:
Setting 1
Setting 2
\begin{tabular}{lrr} 
String & kr 35648,00 & \(\$ 35,648.00\) \\
Number & 35648.00 & 35648.00
\end{tabular}
money ( A, '\#,\#\#0 ¥', '.' , ',' ) where A=3564800 returns:
\begin{tabular}{lrr} 
& Setting 1 & Setting 2 \\
String & \(3,564,800 ¥\) & \(3,564,800 ¥\) \\
Number & 3564800 & 3564800
\end{tabular}

Back to Other Functions.

\section*{Date}

\section*{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.

\section*{Examples:}

The examples below assume the two following operating system settings:

Default setting 1 Default setting 2
Date format YY-MM-DD M/D/YY
date ( A ) where \(A=35648\) returns:
\begin{tabular}{lrr} 
& Setting 1 & Setting 2 \\
String & \(97-08-06\) & \(8 / 6 / 97\) \\
Number & 35648 & 35648
\end{tabular}
date ( A, 'YY.MM.DD' ) where \(A=35648\) returns:
\begin{tabular}{lrr} 
& Setting 1 & Setting 2 \\
String & \(97-08-06\) & \(97-08-06\) \\
Number & 35648 & 35648
\end{tabular}
date ( A, 'DD.MM.YY' ) where \(\mathrm{A}=35648.375\) returns:

Setting 1
\begin{tabular}{lcc} 
String & 06.08 .1997 & 06.08 .1997 \\
Number & 35648.375 & 35648.375
\end{tabular}

Setting 1
Setting 2
\begin{tabular}{llr} 
String & NULL(nothing) & 97.08 .06 \\
Number & NULL & 35648
\end{tabular}

Back to Other Functions

\section*{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.

\section*{Examples:}

The examples below assume the two following operating system settings:
\[
\text { Default setting } 1 \quad \text { Default setting } 2
\]

Time format hh:mm:ss hh.mm.ss
time( \(\mathbf{A}\) ) where \(\mathrm{A}=0.375\) returns:
\begin{tabular}{lrr} 
& Setting 1 & Setting 2 \\
String & \(09: 00: 00\) & 09.00 .00 \\
Number & 0.375 & 0.375
\end{tabular}
time( A ) where \(\mathrm{A}=35648.375\) returns:
\begin{tabular}{lrr} 
& Setting 1 & Setting 2 \\
String & \(09: 00: 00\) & 09.00 .00 \\
Number & 35648.375 & 35648.375
\end{tabular}
time( \(\mathbf{A}, \mathbf{\prime h h}-\mathbf{m m}\) ' ) where \(\mathrm{A}=0.99999\) returns:
Setting 1
\begin{tabular}{lrr} 
String & \(23-59\) & \(23-59\) \\
Number & 0.99999 & 0.99999
\end{tabular}

Back to Other Functions

\section*{Dual}

\section*{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. 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.

\section*{Example:}
```

load dual ( string,numrep ) as DayOfWeek inline
[ string,numrep
Monday,0
Tuesday,1
Wednesday,2
Thursday,3
Friday,4
Saturday,5
Sunday,6 ];
load Date, weekday(Date) as DayOfWeek from afile.csv;

```

The script example will generate a field DayOfWeek with the weekdays written in clear text. QlikView will for all purposes regard the field as a numeric field.

Back to Other Functions.

\section*{Interval}

\section*{interval (expression [ , format-code ])}

The interval function formats the expression as a time interval according to the string given as a formatcode. 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.

\section*{Examples:}

The examples below assume the following operating system settings:
\begin{tabular}{ll} 
Short date format: & YY-MM-DD \\
Time format: & hh:mm:ss
\end{tabular}

\section*{Number decimal separator:}
interval( A ) where \(\mathrm{A}=0.375\) returns:
\begin{tabular}{lr} 
String & \(09: 00: 00\) \\
Number & 0.375
\end{tabular}
interval( A ) where \(\mathrm{A}=1.375\) returns:
\begin{tabular}{lr} 
String & \(33: 00: 00\) \\
Number & 1.375
\end{tabular}
interval( A, 'D hh:mm' ) where \(\mathrm{A}=1.375\) returns:
\begin{tabular}{lr} 
String & \(109: 00\) \\
Number & 1.375
\end{tabular}
interval( A-B, 'D hh:mm' ) where \(A=97-08-06\) 09:00:00 and \(B=96-08-06\) 00:00:00 returns:
\begin{tabular}{lr} 
String & \(36509: 00\) \\
Number & 365.375
\end{tabular}

Back to Other Functions.

\section*{Timestamp}

\section*{timestamp(expression [ , format-code ])}

The timestamp function formats the expression as a date and time according to the string given as formatcode. If the format code is omitted, the date and time formats set in the operating system are used.

\section*{Examples:}

The examples below assume the two following operating system settings:
Default setting \(1 \quad\) Default setting 2
\begin{tabular}{lcc} 
Date format & YY-MM-DD & M/D/YY \\
Time format & hh:mm:ss & hh:mm:ss
\end{tabular}
timestamp( A ) where \(\mathrm{A}=35648.375\) returns:

Setting 1
Setting 2
String 97-08-06 09:00:00
Number
35648.375

8/6/97 09:00:00
35648.375
timestamp( A,'YYYY-MM-DD hh.mm') where A=35648 returns:

\section*{Setting 1}

Setting 2
String
1997-08-06 00.00
1997-08-06 00.00
35648
35648

\section*{Back to Other Functions.}

\section*{Color Functions}

These functions can be used in color expressions in sheet object properties supporting calculated color in image charts.
The functions RBG, 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.

\section*{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 ' \(\operatorname{RGB}(\mathrm{r}, \mathrm{g}, \mathrm{b}\) )' where \(\mathrm{r}, \mathrm{g}\) 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 631) dialog, the function will always return black.

\section*{RGB (e1, e2, e3)}

This function returns the color representation of a color defined by the red component \(e l\), the green component \(e 2\) and the blue component \(e 3\). 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 \(\operatorname{RGB}(r, g, b)\) ' 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.

\section*{ARGB (alpha, e1, e2, e3)}

This function returns the color representation of a color defined by the red component \(e 1\), the green component \(e 2\) and the blue component \(e 3\) 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 ' \(\operatorname{RGB}(a, r, g, b)\) ' 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.

\section*{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, \(g, b)^{\prime}\) 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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{qliktechgray( )}

Returns the RGB color representation for QT gray (RGB 158,148,137).

\section*{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 \(0<\) 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.

\section*{Example:}
```

colormix1(x, black( ) , red( ) )

```

\section*{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 \(-1<\) 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.

\section*{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 param-
eter to the Windows API function GetSysColor(nr). Some values for nr are :
0}\mathrm{ 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}\mathrm{ corresponds to COLOR_ACTIVEBORDER
11}\mathrm{ corresponds to COLOR_INACTIVEBORDER
12 corresponds to COLOR_APPWORKSPACE
13}\mathrm{ corresponds to COLOR_HIGHLIGHT
14 corresponds to COLOR_HIGHLIGHTTEXT
15}\mathrm{ corresponds to COLOR_BTNFACE
16}\mathrm{ corresponds to COLOR_BTNSHADOW
17 corresponds to COLOR_GRAYTEXT
18}\mathrm{ corresponds to COLOR_BTNTEXT
19 corresponds to COLOR_INACTIVECAPTIONTEXT
20}\mathrm{ corresponds to COLOR_BTNHIGHLIGHT
21 corresponds to COLOR_3DDKSHADOW
22 corresponds to COLOR_3DLIGHT
23}\mathrm{ corresponds to COLOR_INFOTEXT
24 corresponds to COLOR_INFOBK
26 corresponds to COLOR_HOTLIGHT (Win2000)
27}\mathrm{ corresponds to COLOR_GRADIENTACTIVECAPTION (Win2000)
28}\mathrm{ corresponds to COLOR_GRADIENTINACTIVECAPTION (Win2000)

```

Back to Other Functions.

\section*{26 Data Structures}

\subsection*{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.

\section*{Rules:}

QlikView does not make any difference between tables generated by a load or a select statement. Hence, if several tables are loaded, it does not matter whether the tables are loaded by load or select statements or by a mix of the two.
The order of the fields in the statement or in the original table in the database is arbitrary to the QlikView logic.
Field names are used in the further process to identify fields and making associations. These are case sensitive. It is thus often necessary to rename fields in the script. See Renaming Fields (page 369).

\subsection*{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

\subsection*{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. The system fields which are typically created during script execution, are primarily used as an aid in document design. The following system fields can be displayed:
\begin{tabular}{ll}
\begin{tabular}{l} 
STable
\end{tabular} & \begin{tabular}{l} 
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.
\end{tabular} \\
\$Field & \begin{tabular}{l} 
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.
\end{tabular} \\
\$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 & \begin{tabular}{l} 
This list box shows the number of rows in the tables.
\end{tabular} \\
\$Info & \begin{tabular}{l} 
If info tables have been included in the document, their names will be dis- \\
played here.
\end{tabular}
\end{tabular}

Tip! A very useful tool is the System table, a pivot table with the two dimensions \$Field and \$Table and the expression only(\$Field). Let the program automatically create this table through the command: Layout: New Sheet Object, System Table.

\subsection*{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:

\section*{concatenate}
this table is added (concatenated) to the last previously created logical table.

\section*{crosstable}
this table is converted from crosstable format to column format.

\section*{generic}
this table is split into several other logical tables.

\section*{info}
this table is loaded not as a logical table, but as an information table containing links to external info such as files, sounds, URLs, etc.

\section*{intervalmatch}
the table (which must contain exactly two columns) is interpreted as numeric intervals, which are associated with discrete numbers in a specified field.

\section*{join}
this table is joined by QlikView with the previously created logical table, over the fields in common.

\section*{mapping}
this table (which must contain exactly two columns) is read as a mapping table, which is never associated with other tables.

\section*{semantic}
this table is loaded not as a logical table, but as a semantic table containing relationships that should not be joined, e.g. predecessor, successor and other references to other objects of the same type.
When the data has been loaded, the logical tables are associated. The logical tables and the associations can be seen in the Table Viewer dialog, see Table Viewer (page 181).

\subsection*{26.5 Associations between Logical Tables}

\section*{Data Associations}

A database can have many tables. Each table can be considered as a list of something, i.e. each record in the list represents an instance of an object of some type.
If two tables are lists of different things, e.g. if one is a list of customers and the other a list of invoices, and the two tables have a field in common, e.g. the customer number, this is usually a sign that there is a relationship between the two tables. In standard SQL query tools the two tables should almost always be joined. The tables defined in the QlikView script are called logical tables. QlikView makes associations between the tables based on the field names, and performs the joins when a selection is made, i.e. when the user clicks on a list box item.
Hence, a QlikView association is almost the same thing as a QlikView join (see below). The only difference is that the join is performed when the script is executed - the logical table is usually the result of the join. The association is made after the logical table is created - associations are always made between the logical tables.


Four tables: a list of countries, a list of customers, a list of transactions and a list of memberships, which are associated with each other through the fields Country and CustomerID.
A QlikView association has approximately the same effect as a SQL natural outer join. The QlikView association is however more general: an outer join in SQL is usually a one way projection of one table on another. A QlikView association always results in a full (bidirectional) natural outer join.

\section*{Frequency Information in Associating Fields}

There are some limitations in the use of most associating fields, i.e. fields which are common between two or more tables. When a field occurs in more than one table, QlikView has a problem knowing which of the tables it should use for calculating data frequencies.

QlikView analyzes the data to see if there is a non-ambiguous way to identify a main table to count in (sometimes there is) but in most cases the program can only make a guess. Since an incorrect guess could be fatal (QlikView would appear to make a calculation error) the program has been designed not to allow certain operations when the data interpretation is ambiguous for associating fields.
For associating fields the following limitations normally apply:
1. 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 \(\mathrm{n} / \mathrm{a}\) 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. Try
and an error message will appear. After each reload, QlikView will scan all chart expressions to see if any ambiguities have occurred as a result of changes in data structures. If ambiguous expressions are found, a warning dialog will be shown and the expression will be disabled. It will not be possible to enable the expression until the problem has been corrected. If a log file is enabled, all ambiguous expressions will be listed in the log.
There is a simple way to overcome these limitations. Load the field an extra time under a new name from the table where frequency counts should be made. Then use the new field for a list box with frequency, for a statistics box or for calculations in the charts.

\section*{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.
Therefore we recommend an overall analysis of the intended table structure by the application designer. Typical tricks include:
- 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.

\section*{Circular References ("Loops")}

If there are circular references ("loops") in a data structure, the tables are associated in such a way that there is more than one path of associations between two fields.
This type of data structure should normally be avoided as much as possible, since it might lead to ambiguities in the interpretation of data. Unfortunately, circular structures are quite common in the real world. In some cases they are a result of poor database design, but in certain cases they may be unavoidable.


\section*{Three tables with a circular reference}

QlikView solves the problem of circular references by breaking the loop with a loosely coupled table. When QlikView finds circular data structures while executing the load script, a warning dialog will be shown and one or more tables will be set as loosely coupled. QlikView will typically attempt to loosen the longest table in the loop, as this is often a transaction table, which normally should be the one to loosen. If you do not like QlikView's default choice of table to set as loosely coupled, it is possible to declare a specific table as loosely coupled via a loosen table statement in the script. It is also possible to change the settings for loosely coupled tables interactively in the Document Properties: Tables page after the script has been executed. See chapter Logic and Selections (page 107), for more information on loosely coupled tables.

\subsection*{26.6 Renaming Fields}

Sometimes it is necessary to rename fields in order to obtain the desired associations.
It could be that two fields are named differently although they denote the same thing, e.g. ID in a table Customers and CustomerID in a table Orders. Obviously they both denote a specific customer identification code and should both be named CustomerID or something similar.
It could also be that two fields are named the same way but actually denote different things, e.g. Date in the table Invoices and Date in the table Orders. These should preferably be renamed InvoiceDate and OrderDate or something similar.
There could also be plain misspellings in the database or different conventions on upper and lower case letters. (Since QlikView differs between upper and lower case letters it is important that corrections are made.)
Fields can be renamed in the script, so there is no need to change the original data. There are two ways to do it:

The Load (page 246) or Select (SQL) (page 262) statement can be preceded by an Alias (page 223) statement.

\section*{Example:}

Alias ID as CustomerID;
Load * from Customer.csv;

The load or select statement can contain the as specifier.

\section*{Example:}

Load ID as CustomerID, Name, Address, Zip, City, State from Cus-
tomer.csv;

Finally, it is also possible to rename list boxes and other sheet objects, thereby changing the labels while keeping the logical associations defined by the field names intact. Choose Properties from the object menu of the sheet object.

\subsection*{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.

\section*{Example:}
```

load a, b, c from table1.csv;
load a, c, b from table2,csv;

```

The resulting internal table has the fields \(\mathrm{a}, \mathrm{b}\) and c . The number of records is the sum of the numbers of records in table 1 and table 2.

\section*{Rules:}
- The number and names of the fields must be exactly the same.
- The order of the two statements is arbitrary.

\section*{Forced Concatenation}

If two or more tables do not have exactly the same set of fields, it is still possible to force QlikView to concatenate the two tables. This is done with the concatenate prefix in the script, which concatenates a table with another named table or with the last previously created table.

\section*{Example:}
```

load a, b, c from tablel.csv;
concatenate load a, c from table2,csv;

```

The resulting internal table has the fields \(\mathrm{a}, \mathrm{b}\) 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.

\section*{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.

\section*{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.

\section*{Example:}
```

load a, b, c from table1.csv;

```
nonconcatenate load \(a, b, c\) from table2.csv

\subsection*{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. Sometimes this is preferable, sometimes even necessary, but most of the time it is not advisable, since:
- The loaded tables often become larger, and QlikView works slower.
- Some information may be lost: the frequency (number of records) within the original table may no longer be available.
The keep functionality, which has the effect of reducing one or both of the two tables to the intersection of table data before the tables are stored in QlikView, has been designed to reduce the number of cases where explicit joins needs to be used. .

\section*{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.

\section*{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.

\section*{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.

\section*{Join}

The simplest way to make a join is with the join prefix in the script, which joins the internal table with the last previously created table. The join will be an outer join, creating all possible combinations of values from the two tables.

\section*{Example:}
```

load a, b, c from tablel.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.

\section*{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.

\section*{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 results 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 needs 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.

\section*{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.

\section*{Example:}

Table1


Table2

```

QVTable:
Select * from Table1;
inner join select * from Table2;

```

\section*{QVTable}

\section*{A B C}


QVTab1:
Select * from Table1;
QVTab2:
inner keep select * from Table2;

QVTab1



The two tables in the keep example are of course associated via the field \(A\).

\section*{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.

\section*{Example:}

\section*{Table1}


Table2

```

QVTable:
Select * from Table1;
left join select * from Table2;

```

QVTable
A B C

```

QVTab1:
Select * from Table1;
QVTab2:
left keep select * from Table2;

```

QVTab1
A B
\begin{tabular}{|l|l|}
\hline 1 & aa \\
\hline 2 & cc \\
\hline 3 & ee \\
\hline
\end{tabular}

QVTab2
A C


The two tables in the keep example are of course associated via the field A.

\section*{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.

\section*{Example:}

\section*{Table1}


Table2

```

QVTable:
Select * from Table1;
right join select * from Table2;

```

\section*{QVTable}

```

QVTab1:
Select * from Table1;
QVTab2:
right keep select * from Table2;

```


The two tables in the keep example are of course associated via the field A.

\section*{27 Evaluating the Loaded Data}

\subsection*{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.
\begin{tabular}{||l|l|l|}
\hline \multicolumn{3}{|l|}{ Generic database } \\
\hline object & attribute & value \\
\hline ball & color & red \\
\hline ball & diameter & 10 cm \\
\hline ball & weight & 100 g \\
\hline box & color & black \\
\hline box & height & 16 cm \\
\hline box & lenght & 20 cm \\
\hline box & weight & 500 g \\
\hline box & width & 10 cm \\
\hline \hline
\end{tabular}

This is a typical generic database. On one hand it would be awkward to store the data in a way giving each attribute a column of its own, since many of the attributes are not relevant for a specific object.
On the other hand, it would look messy displaying it in a way that mixed lengths, colors and weights.
\begin{tabular}{|c|c|c|}
\hline object & Ettribute & value \\
\hline ball & color & black \\
\hline pox & diameter & red \\
\hline & height & 10 cm \\
\hline & length & 16 cm \\
\hline & weight & 20 cm \\
\hline & width & 100 g \\
\hline & & 500 g \\
\hline
\end{tabular}

QlikView solves this problem in an elegant way. If data is stored in the compact way shown above, choose between two different ways of displaying the data. QlikView automatically creates several internal tables from the generic database.
If this table is loaded the standard way, we can get three different list boxes on the screen. However, if the table is loaded as a generic database, QlikView will split up column two and three into different list boxes. QlikView will then generate one field for each unique value of the second column.


The syntax for doing this is simple:

\section*{Example:}

Generic select * from GenericTable;
It does not matter whether a load or select statement is used to load the generic database.

\subsection*{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.

\section*{Example 1:}
ex1.csv
\begin{tabular}{ccccccc} 
Year & Jan & Feb & Mar & Apr & May & Jun \\
1991 & 45 & 65 & 78 & 12 & 78 & 22 \\
1992 & 11 & 23 & 22 & 22 & 45 & 85 \\
1993 & 65 & 56 & 22 & 79 & 12 & 56 \\
1994 & 45 & 24 & 32 & 78 & 55 & 15 \\
1995 & 45 & 56 & 35 & 78 & 68 & 82
\end{tabular}

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 exl.csv;

```

The result in QlikView would be as follows:
\begin{tabular}{|c|c|c|c|c|}
\hline Year & Month & \multicolumn{3}{|l|}{Sales} \\
\hline 1991 & Apr & 11 & 45 & 85 \\
\hline 1992 & Feb & 12 & 55 & \\
\hline 1993 & Jan & 15 & 56 & \\
\hline 1994 & Jun & 22 & 65 & \\
\hline 1995 & Mar & 23 & 68 & \\
\hline & May & 24 & 78 & \\
\hline & & 32 & 79 & \\
\hline & & 35 & 82 & \\
\hline
\end{tabular}

The cross table is often preceded by a number of qualifying columns, which should be read in a straightforward way. This is the case in example 2 :

\section*{Example 2:}
ex2.csv
\begin{tabular}{lccccccc} 
Salesman & Year & Jan & Feb & Mar & Apr & May & Jun \\
A & 1991 & 45 & 65 & 78 & 12 & 78 & 22 \\
A & 1992 & 11 & 23 & 22 & 22 & 45 & 85
\end{tabular}
\begin{tabular}{llllllll} 
A & 1993 & 65 & 56 & 22 & 79 & 12 & 56 \\
A & 1994 & 45 & 24 & 32 & 78 & 55 & 15 \\
A & 1995 & 45 & 56 & 35 & 78 & 68 & 82 \\
B & 1991 & 57 & 77 & 90 & 24 & 90 & 34 \\
B & 1992 & 23 & 35 & 34 & 34 & 57 & 97 \\
B & 1993 & 77 & 68 & 34 & 91 & 24 & 68 \\
B & 1994 & 57 & 36 & 44 & 90 & 67 & 27 \\
B & 1995 & 57 & 68 & 47 & 90 & 80 & 94
\end{tabular}

In this case there are two qualifying columns to the left, followed by the matrix columns. The number of qualifying columns can be stated as a third parameter to the crosstable prefix as follows:
```

crosstable (Month, Sales, 2) load * from ex2.csv ;

```

The result in QlikView would be:
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline Salesman & Year & Month & \multicolumn{4}{|l|}{Sales} \\
\hline A & 1991 & Apr & 11 & 34 & 57 & 82 \\
\hline B & 1992 & Feb & 12 & 35 & 65 & 85 \\
\hline & 1993 & Jan & 15 & 36 & 67 & 90 \\
\hline & 1994 & Jun & 22 & 44 & 68 & 91 \\
\hline & 1995 & Mar & 23 & 45 & 77 & 94 \\
\hline & & May & 24 & 47 & 78 & 97 \\
\hline & & & 27 & 55 & 79 & \\
\hline & & & 32 & 56 & 80 & \\
\hline
\end{tabular}

For a description of the syntax see Crosstable (page 229).

\subsection*{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.

\section*{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?

\section*{OrderLog:}
\begin{tabular}{|cll|}
\hline START & END & ORDER \\
\hline \(\mathbf{0 1 : 0 0}\) & \(\mathbf{0 3 : 3 5}\) & A \\
\(\mathbf{0 2 : 3 0}\) & \(\mathbf{0 7 : 5 8}\) & B \\
\(\mathbf{0 3 : 0 4}\) & \(10: 27\) & C \\
\(\mathbf{0 7 : 2 3}\) & \(11: 43\) & D \\
\hline
\end{tabular}

\section*{EventLog:}

\section*{Time Event Comment}
\begin{tabular}{|lll|}
\hline \(00: 00\) & 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 \\
\hline
\end{tabular}

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:
\begin{tabular}{|c|c|c|c|c|}
\hline & & \multicolumn{3}{|r|}{昌 XL - \(\square\)} \\
\hline Time & Event Comment & Order & Start & End \\
\hline 00:00 & 0 Start of shift 1 & - & - & \\
\hline 01:18 & 1 Line stop & A & 01:00 & 03:35 \\
\hline 02:23 & 2 Line restart 50\% & A & 01:00 & 03:35 \\
\hline 04:15 & 3 Line speed 100\% & 日 & 02:30 & 07:58 \\
\hline 04:15 & 3 Line speed 100\% & C & 03:04 & 10:27 \\
\hline 08:00 & 4 Start of shift 2 & C & 03:04 & 10:27 \\
\hline 08:00 & 4 Start of shift 2 & D & 07:23 & 11:43 \\
\hline 11:43 & 5 End of production & D & 07:23 & 11:43 \\
\hline
\end{tabular}

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.

\subsection*{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.

\section*{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;

```

\section*{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:
\begin{tabular}{|r|l|l|}
\hline Table1 \\
\hline Key & FirstDate & Team \\
\hline 000110 & & Northwest \\
\hline 000110 & \(2001-01-21\) & Southwest \\
\hline 000120 & & Northwest \\
\hline 000120 & \(2003-01-06\) & Southeast \\
\hline 000120 & \(2003-03-05\) & Northwest \\
\hline
\end{tabular}

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:
\begin{tabular}{|l|l|r|l|}
\hline Table2 2 & \multicolumn{4}{l|}{} \\
\hline Key & Name & \multicolumn{1}{l|}{ Date } & Sales \\
\hline 0000110 & Spengler Aaron & \(1999-08-18\) & 100 \\
\hline 000110 & Spengler Aaron & \(1999-12-25\) & 200 \\
\hline 000110 & Spengler Aaron & \(2001-02-03\) & 300 \\
\hline 000110 & Spengler Aaron & \(2001-05-05\) & 400 \\
\hline 000120 & Ballard John & \(2001-06-04\) & 500 \\
\hline 000120 & Ballard John & \(2003-01-20\) & 600 \\
\hline 000120 & Ballard John & \(2003-03-10\) & 700 \\
\hline 000120 & Ballard John & \(2003-03-1\) & 800 \\
\hline 000120 & Ballard John & \(2003-09-21\) & 900 \\
\hline
\end{tabular}

The intervalmatch statement preceeded by the inner join replaces the key above with a synthetic key that connects to the Transact table resulting in the following table:
\begin{tabular}{|r|l|l|l|r|r|r|}
\hline \multicolumn{7}{|l|}{ Table3 } \\
\hline Key & Team & Name & FirstDate & \multicolumn{1}{l|}{ LastDate } & TransactDate & Sales \\
\hline 000110 & Northwest & Spengler Aaron & & \(2001-01-20\) & \(1999-08-18\) & 100 \\
\hline 000110 & Northwest & Spengler Aaron & & \(2001-01-20\) & \(1999-12-25\) & 200 \\
\hline 000110 & Southwest & Spengler Aaron & \(2001-01-21\) & & \(2001-02-03\) & 300 \\
\hline 000110 & Southwest & Spengler Aaron & \(2001-01-21\) & & \(2001-05-05\) & 400 \\
\hline 000120 & Northwest & Ballard John & & \(2003-01-05\) & \(2001-06-04\) & 500 \\
\hline 000120 & Northwest & Ballard John & \(2003-03-05\) & & \(2003-03-10\) & 700 \\
\hline 000120 & Northwest & Ballard John & \(2003-03-05\) & & \(2003-03-13\) & 800 \\
\hline 000120 & Northwest & Ballard John & \(2003-03-05\) & & \(2003-09-21\) & 900 \\
\hline 000120 & Southeast & Ballard John & \(2003-01-06\) & \(2003-03-04\) & \(2003-01-20\) & 600 \\
\hline
\end{tabular}

\subsection*{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.
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 236) for more information.
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline NodelD & ParentID & NodeName & Level0 & Level1 & Level2 & Level3 & Level4 & Level5 \\
\hline 1 & - & The World & The World & - & - & - & - & - \\
\hline 2 & 1 & Europe & The World & Europe & - & - & - & - \\
\hline 3 & & France & The World & Europe & France & - & - & - \\
\hline 4 & F & Bordeaux & The World & Europe & France & Bordeaux & - & - \\
\hline 5 & & Medoc & The World & Europe & France & Bordeaux & Medoc & - \\
\hline 6 & 5 & Bas-Médoc & The World & Europe & France & Bordeaux & Medoc & Bas-Médoc \\
\hline 7 & 5 & Haut-Médoc & The World & Europe & France & Bordeaux & Medoc & Haut-Médoc \\
\hline 8 & 4 & Graves & The World & Europe & France & Bordeaux & Graves & - \\
\hline 9 & 3 & Bourgogne & The World & Europe & France & Bourgogne & - & - \\
\hline 10 & 2 & Germany & The World & Europe & Germany & - & - & - \\
\hline 11 & 10 & Rheingau & The World & Europe & Germany & Rheingau & - & - \\
\hline 12 & 1 & Americas & The World & Americas & - & - & - & - \\
\hline 13 & 12 & California & The World & Americas & California & - & - & - \\
\hline 14 & 13 & Napa valley & The World & Americas & California & Napa valley & - & - \\
\hline
\end{tabular}

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.
\begin{tabular}{|c|c|c|c|}
\hline NodelD & NodeName & AncestoriD & AncestorName \\
\hline 1 & The World & & The World \\
\hline & 2 Europe & 1 & The World \\
\hline & 2 Europe & & 2 Europe \\
\hline & France & 1 & The World \\
\hline & France & & Europe \\
\hline & France & & France \\
\hline & Bordeaux & & The World \\
\hline & Bordeaux & & Europe \\
\hline & Bordeaux & & France \\
\hline & Bordeaux & & Bordeaux \\
\hline & 5 Medoc & 1 & The World \\
\hline & 5 Medoc & & Europe \\
\hline & 5 Medoc & & France \\
\hline & Medoc & & Bordeaux \\
\hline
\end{tabular}

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 214) 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.


\subsection*{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.

\section*{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.

\section*{Relation}
s- 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.

\section*{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.

\section*{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.
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|l|}{Semantic table} \\
\hline No & Relation & No & Relation \\
\hline 1 & Successor & 2 & Predecessor * \\
\hline 2 & Successor & 3 & Predecessor \\
\hline 3 & Successor & 4 & Predecessor \\
\hline 4 & Successor & 5 & Predecessor \\
\hline 5 & Successor & 6 & Predecessor \\
\hline 6 & Successor & 7 & Predecessor \\
\hline 7 & Successor & 8 & Predecessor \\
\hline 8 & Successor & 9 & Predecessor \\
\hline 9 & Successor & 10 & Predecessor \\
\hline 10 & Successor & 11 & Predecessor \\
\hline
\end{tabular}

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.

\section*{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 Dummyl 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.

\subsection*{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 QlikView 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 " \(\backslash \mathrm{n}\) " 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 388), however, is allowed.

Using bundle info load, it is possible to bundle external data into a QlikView document. Read more about Bundle (page 224).

\subsection*{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.

\section*{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

\section*{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.

\section*{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 253)) 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 254)) 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 369)), include the field Country, whose field values will be compared with those of the first column of the mapping table. The field values U.S., United States and US will be found and replaced by the values of the second column of the mapping table, i.e. USA.
The automatic mapping is done last in the chain of events that leads up to the field being stored in the QlikView table. For a typical load or select statement the order of events is roughly as follows:
1. Evaluation of expressions
2. Renaming of fields by as
3. Renaming of fields by alias
4. Qualification of table name, if applicable
5. Mapping of data if field name matches

This means that the mapping is not done every time a field name is encountered as part of an expression but rather when the value is stored under the field name in the QlikView table.
To disable mapping, use the unmap statement (see Unmap (page 271)).
For mapping on expression level, use the applymap function (see Mapping Functions (page 325)).
For mapping on substring level, use the mapsubstring function (see Mapping Functions (page 325)).

\subsection*{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.

\section*{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 266).

\section*{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=;

\section*{Example:}

\section*{Customers}


\section*{Orders}
\begin{tabular}{|l|c|} 
CustomerID & OrderID \\
\hline 1 & 1234 \\
\hline 3 & 1243 \\
\hline 5 & 1248 \\
\hline 7 & 1299 \\
\hline
\end{tabular}

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.

\section*{Note!}

OtherSymbol is not intended to be used for creating outer joins between tables!!

\subsection*{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 255). Furthermore, NULL values from ODBC, can be manipulated using nulldisplay (see below).
Text files per definition cannot contain NULL values.

\section*{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=;

\section*{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.

\section*{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=;

\section*{Note!}

The use of NULLINTERPRET only affects data from text files and inline clauses!

\section*{Propagation of NULL Values in Expressions}

NULL values will propagate through an expression according to a few logical and quite reasonable rules.

\section*{Functions}

The general rule is that functions return NULL when the parameters fall outside the range for which the function is defined.

\section*{Examples:}
\begin{tabular}{lll}
\(\operatorname{asin}(2)\) & returns & NULL \\
\(\log (-5)\) & returns & NULL \\
round(A,0) & returns & NULL
\end{tabular}

As a result of the above follows that functions generally return NULL when any of the parameters necessary for the evaluation are NULL.

\section*{Examples:}
\begin{tabular}{lll}
\(\sin (\) 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
\end{tabular}

The exception to the second rule are logical functions testing for type.

\section*{Examples:}
```

isnull(NULL) returns TRUE (-1)
isnum(NULL) returns FALSE (0)

```

\section*{Arithmetic and String Operators}

If NULL is encountered on any side of these operators NULL is returned, except in the case of string concatenation.

\section*{Examples:}
\begin{tabular}{lll} 
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
\end{tabular}

\section*{Relational Operators}

If NULL is encountered on any side of relational operators, special rules apply.

\section*{Examples:}
\begin{tabular}{lll} 
NULL rel.op. NULL & returns & NULL \\
A \(\diamond\) NULL & returns & TRUE (-1) \\
& & \\
A \(<\) NULL & returns & FALSE (0)
\end{tabular}
\begin{tabular}{lll}
\(\mathrm{A}<=\) NULL & returns & FALSE (0) \\
\(\mathrm{A}=\) NULL & returns & FALSE (0) \\
\(\mathrm{A}>=\) NULL & returns & FALSE (0) \\
\(\mathrm{A}>\) NULL & returns & FALSE (0)
\end{tabular}

\subsection*{27.11 Character Set}

Available character sets are:
ansi
oem
mac
UTF-8
Unicode
The file can be written with the ansi character set (Windows), with the oem character set (DOS, OS/2 and others), Unicode, UTF-8 or with the mac. The conversion from the oem character set is not implemented for MacOS. If nothing is specified, ansi is assumed under Windows.

\section*{Example:}
load * from a.txt (CodePage is 10000, txt, delimiter is ',' , no
labels)

Back to Script Statements and Keywords (page 219).

\section*{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 super-fast. The selected mode is determined automatically by the QlikView script engine. Super-fast mode can be utilized only when all fields or a subset thereof are read without any transformations (formulas acting upon the fields), though the renaming of fields is allowed.

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.

\subsection*{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:

\section*{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.

\section*{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. The other applications can make use of the same data via a QVD file.

\section*{Consolidating Data from Multiple QlikView Applications}

With the Binary (page 223) script statement it is limited to loading data from a 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.

\section*{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 394)

\subsection*{28.2 Create QVD Files}

A QVD file can be created by one of three different methods:
1. Explicit creation and naming from script by means of the Store (page 267) command. 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 873). 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.

\subsection*{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 196) handles QVD files by the same principles.

\section*{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 334) functions.

\subsection*{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 393) 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 395) (Log files).
2) Case 2: Insert Only (No Update or Delete) (page 396) (No Update or Delete).
3) Case 3: Insert and Update (No Delete) (page 397) (No Delete).
4) Case 4: Insert, Update and Delete (page 398).

Below you will find outlined solutions for each of these cases. The reading of QVD files can be done in either "super-fast mode" or "standard mode". (The method employed is automatically selected by the QlikView script engine depending on the complexity of the operation.) "super-fast mode" is (very approximately) about 10x faster than "standard mode" or about 100x faster than loading the database in the ordinary fashion.

\section*{Case 1: Append Only}

The simplest case is the one of log files, which are only appended. The following conditions apply:
- The database must be a log file 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.


\section*{Script Example:}

Buffer (Incremental) Load * From LogFile.txt (ansi, txt, delimiter is '\t', embedded labels);

\section*{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.


\section*{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;

```

\section*{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 "super-fast mode"), which is still considerably faster than loading the entire database.


\section*{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;

```

\section*{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 "super-fast mode"), which is still considerably faster than loading the entire database.


\section*{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

```

\section*{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.

\subsection*{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.

\subsection*{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.

\subsection*{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.

\subsection*{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.

\section*{Example:}
```

Section Access;
Load * inline
[ACCESS,USERID,PASSWORD
ADMIN, A,X
USER,U,Y ];
Section Application;
Load... ... from... ...

```

\subsection*{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.

\section*{Note!}

ADMIN rights are only relevant for local documents! Documents opened on a Server are always accessed with USER rights.

\subsection*{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.
\begin{tabular}{ll} 
ACCESS & A field that defines what access the corresponding user should have. \\
USERID & 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. \\
PASSWORD & 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.
\end{tabular} SERIAL \(\quad\)\begin{tabular}{l} 
A field that should contain a number corresponding to the QlikView serial number. \\
Example: 490023947113 7304 \\
QlikView will check the serial number of the user and compare it to the value in this \\
field.
\end{tabular}
NTNAME
NTDOMAINSID \begin{tabular}{l} 
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. \\
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.
\end{tabular}
NTSID
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.

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 322), 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.

\section*{Note!}

When loading data from a QVD file, the use of the upper function will slow down the loading speed.

\footnotetext{
Note!
To generate access tables in inline statements use the Access Restriction Table Wizard (page 193).
}

\section*{Example 1:}

Only serial number is checked. One specific computer gets ADMIN access. Everyone else gets USER access. Note that a star can be used to mark "any serial number". For this, however, the statement "Star is *;" must precede the loading of this table.

\section*{ACCESS SERIAL}
```

ADMIN 4900 239471137304
USER *

```

\section*{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.
\begin{tabular}{lllll} 
ACCESS & SERIAL & NTDOMAINSID & USERID & PASSWORD \\
ADMIN & \(*\) & S-1-5-21-125976590-467238106- & ADMIN & ADMIN \\
& & 1092489882 & \(*\) & \(*\) \\
ADMIN & 490023947113 & \(*\) & & \\
& 7304 & & \\
USER & \(*\) & S-1-5-21-125976590-467238106- & USER & USER \\
& & 1092489882 & &
\end{tabular}

\subsection*{29.7 Restrictions on QlikView Functionality}

The controls found on the Document Properties: Security (page 423) page and the Sheet Properties: Security (page 451) 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.

\subsection*{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.

\section*{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.

\section*{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.

\subsection*{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.

\subsection*{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.

\section*{Part 5 Sheet and Sheet Objects}

\section*{30 Document Properties}

Opens the Document Properties: General (page 408) page. Here the document specific parameters, such as opening picture and sound, default font, sort order etc. can be specified for this specific document.

\subsection*{30.1 Document Properties: General}


Document Properties, General

Title

Author
Save Format

Alert Pop-ups
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.

In the Author edit box specify the author of the QlikView file.
In this group set the level of compression for the QlikView files.

\section*{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 81) in the User Preferences dialog.

This button opens the Pop-up Window Settings dialog where the default look of alert pop-up messages can be determined.

Help Pop-ups

Alternate States...

Memory Statistics

Default Sheet Background

Use Passive FTP Semantics

Generate Logfile To track the execution of the load script select Generate Logfile. The information that is shown in the Progress window during script execution will be stored as qv.log.
\begin{tabular}{|c|c|}
\hline Timestamp in Logfile Name & Puts a timestamp in the logfile name, e.g. sales.qvw.2009_02_26_12_09_ 50.log. The setting is only available if the Generate Logfile check-box is marked. \\
\hline \begin{tabular}{l}
Hide Unavailable \\
Menu Options
\end{tabular} & The setting Hide Unavailable Menu Options 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. \\
\hline Hide Tabrow & The QlikView sheet tabrow may be removed by checking Hide Tabrow. In order to move between sheets use the Actions (page 532) (Function: Next Sheet / Previous Sheet) or the Sheet Toolbar of the View Menu (page 58). 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. \\
\hline Keep Unreferenced QVD Buffers & The Keep Unreferenced QVD Buffers setting overrules the normal procedure whereby any automatically created QVD Files (page 393) 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. \\
\hline Legacy Fractile Calculation & If the Legacy Fractile Calculation 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. \\
\hline Disable Layout Undo & When the check box Disable Layout Undo 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. \\
\hline 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. \\
\hline Default Export Encoding & Use Default Export Encoding to set the default character set for export in new documents. Choose one of these option: ANSI, Unicode or UTF-8. \\
\hline Styling Mode & In the drop-down menu Styling Mode choose the object style for all the sheet objects. \\
\hline & The Advanced mode makes it possible to configure several settings, such as the Sheet Object Style and the type of border to be used, set on the Layout page of the objects. \\
\hline & The Simplified mode also gives the choice of Sheet Object Style, but configures most settings automatically. It sets such things as Scrollbar Style and borders. Some settings can still be changed on the Layout (page 476) page of the objects. \\
\hline Sheet Object Style & In the drop-down Sheet Object Style select a style for sheet object captions. The style selected will be used for all sheet objects with a caption in the document. \\
\hline
\end{tabular}

\section*{Timestamp in Logfile}

Hide Unavailable Menu Options

Hide Tabrow

\section*{Unreferenced QVD Buffers}

\section*{Legacy Fractile Calculation}

\section*{Disable Layout Undo}

Use WebView in Layout

Default Export Encoding

Styling Mode

Sheet Object Style

In the drop-down Sheet Object Style select a style for sheet object captions. ument.

Tabrow Style

\section*{Selection Appearance}

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.

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 77) and List Box Properties: Presentation (page 471). 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.

\section*{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.

\section*{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.

\section*{Transparency}

Sets the transparency of the selection color in list boxes and multi boxes.
Specify a custom color to the background of the tab row by clicking the corresponding Tabrow Background button.

\section*{Pop-Up Window Settings}


\footnotetext{
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 475) 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 412) 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 385).
The edit box below should contain a valid path to a wav sound file. This may be an external path (e.g. c: \(\backslash-\) mysound.wav) or a path to a QlikView sound (Internal Files (page 165)) already bundled with the program (e.g. qmem://<bundled>/sounds/qv_ok.wav).

\section*{Color Area}


\section*{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.

\section*{Solid Color}

One Color Gradient This option introduces the use of varying Brightness to create a color gradient effect.

Two Color 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.
Fixed This alternative specifies a single, basic color. Click the colored button to change the color.

Calculated The 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 361). 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.

Brightness When using a One Color Gradient 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.
Fixed This alternative specifies a single, basic color. Click the colored button to change the color.

Calculated The color may be dynamically calculated from an expression, using the Color Functions (page 361).

In the Shading Style group the direction of the color gradient effect is determined. It is possible to achieve different results by combining the available alternatives (Horizontal, Vertical etc.) with any of the four Variants.

The Variants group presents four different alternatives for each Shading Style selected. Click the preferred variant.

\subsection*{30.2 Document Properties: Opening}


\section*{Document Properties, Opening}

On this tab it is possible to customize the opening of a document.
\(\left.\begin{array}{ll}\text { Image } & \begin{array}{l}\text { Mark this check box to have an opening image for the document. } \\
\text { Select... } \\
\text { Close on Mouse Click } \\
\text { Click this button to open the Select Image dialog where it is possible to select } \\
\text { an image (bmp, jpg, jpeg or png). }\end{array} \\
\text { Close after N Seconds } & \begin{array}{l}\text { Lets the user close the opening image by clicking it. Either this check box or } \\
\text { the one below must be marked. }\end{array} \\
\text { Closes the opening image automatically. Either this check box or the one } \\
\text { above must be marked. }\end{array}\right]\)\begin{tabular}{l} 
Closes the opening image when the file has been loaded. \\
Sound \\
Select...
\end{tabular}\(\quad\)\begin{tabular}{l} 
Mark this check box to add an opening sound for the document. This makes \\
the Select button available.
\end{tabular}
\begin{tabular}{ll} 
Initial Data Reduction & \begin{tabular}{l} 
If the Security (page 401) option is selected, QlikView will perform a pro- \\
cedure on opening the document (or on establishing a new session on Qlik- \\
Based on Section \\
Access
\end{tabular} \\
\begin{tabular}{l} 
View Server) which permanently hides certain data from the user based on the \\
identity with which the user has logged on to the document.
\end{tabular} \\
& \begin{tabular}{l} 
When this feature is used on a document not run exclusively on a QlikView \\
Server it should always be combined with the Prohibit Binary Load option \\
(see below) in order to maintain data access security. This setting is mutually \\
exclusive with Initial Selection Based on Section Access.
\end{tabular} \\
Strict Exclusion & \begin{tabular}{l} 
If this check box is marked in combination with Initial Data Reduction Based \\
on Section Access, strict exclusion will be used when reducing the data. This \\
means that access to the document will be denied whenever the field values in \\
the section access reduction fields lack matches in their corresponding section
\end{tabular} \\
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 set- \\
ting is enabled by default in documents created with QlikView 7.02 and later \\
and is recommended for maximum access security.
\end{tabular}

\subsection*{30.3 Document Properties: Sheets}


\section*{Document Properties, Sheets}

The Document Properties: Sheets tab helps to keep track of every sheet and sheet object of the QlikViewdocument. 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:
\begin{tabular}{ll} 
\# & 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.
\end{tabular}

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:
\begin{tabular}{ll} 
SheetID & The unique ID of the sheet. \\
ObjectID & Lists the unique Object ID of every existing sheet object. Linked objects have \\
the same ID.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Type & Clarifies the object type. \\
\hline Caption & The text currently shown in the caption of the sheet object. \\
\hline ShowMode & Indicates whether the object is normal, minimized, maximized, hidden or has a show condition. \\
\hline CalcTime & 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). \\
\hline Layer & The layer as defined on the Layout page of a sheet object properties dialog, defined as Bottom, Normal or Top 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.) \\
\hline Memory & 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). \\
\hline Left, Top, Right, Bottom & The current position of the indicated edge of the sheet object relative to the edge of the document window, expressed in pixels. \\
\hline Width, Height & The size dimensions of the object, expressed in pixels. \\
\hline \begin{tabular}{l}
Left (Min), Top (Min), \\
Right (Min), \\
Bottom (Min)
\end{tabular} & The current position of the object when iconized (=minimized), expressed in pixels. \\
\hline Width & The current width of the sheet object's minimized icon, expressed in pixels. \\
\hline Height & The current height of the sheet object's minimized icon, expressed in pixels. \\
\hline Export Structure & 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 (filename.Sheets.tab), and one for the sheet objects (filename.Objects.tab) can easily be read back into QlikView for further analysis with the full power of the QlikView logic. A dialog will appear where the target folder for the export can be chosen. The default is to put the files in the same folder as the QlikView document. \\
\hline \multicolumn{2}{|l|}{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.} \\
\hline
\end{tabular}

\subsection*{30.4 Document Properties: Server}


\section*{Document Properties, Server}

On this tab define certain aspects of the document's behavior when run on Q QlikView Server.

\section*{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.

Client initiates refresh. If old data not kept in server or client too old, refresh will be performed automatically

\section*{Server performs} refresh automatically, without client action

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.

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.

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.

\section*{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).

\section*{Indicate with toolbar button}

\section*{Prompt with dialog}

Just do it but tell the user it is happening

\section*{Just do it}

The availability of new data is indicated by the Refresh button in the toolbar turning green and being enabled (and the corresponding File menu command being enabled). The refresh is performed at a time of the user's choice by clicking on the toolbar button or by means of the File menu Refresh command.

In addition to the Refresh button in the toolbar turning green, the availability of new data will be indicated by a prompt dialog. In the dialog the user may choose between immediate refresh or wait, in which case the refresh can be performed at any later time of the user's choice by clicking on the toolbar button or by means of the File menu Refresh command.

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.

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.

\section*{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.

\section*{Allow Server Bookmarks}

\section*{Allow Server Objects}

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.

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.

\section*{Allow Server Reports}

This check box must be enabled, if remote clients are to be allowed to create and share reports with this document on the QlikView Server.

\section*{OEM Product ID}

The settings in this group make it possible to add an OEM Product ID to the document, for more information see Add OEM Product ID. This option is only available with an OEM Partner License.

OEM Product ID Select OEM Product ID from the drop-down list.

\section*{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.

\section*{Maximum Inactive Session Time (seconds)}

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.

Maximum Total Session Time (seconds)

Enable Push from Server

Enable Dynamic Data Update

Filter AccessPoint Document List Based on Section Access

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.

Mark this check box if the Server should allow graceful document refresh.

Mark this check box if the Server should allow dynamic updates in a document.

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 quw file, however, no passwords will be saved.

\subsection*{30.5 Document Properties: Scheduler}


\section*{Document Properties, Scheduler}

On this tab a default reload schedule for the document in QlikView Server can be proposed.

\section*{Preferred Schedule}

In this drop-down menu set the schedule for how often the document should be reloaded. The schedule can be set to None, Run Once, Hourly, Daily, Weekly and Monthly. Then set the details of the schedule, such as start time, days of the week etc, in the fields below.

\subsection*{30.6 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 171) that have been added to Displayed Variables in the Input Box Properties: General (page 521) 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 737).
Variables are normally not included in bookmarks, but here it is possible to check the option Include in

\section*{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

\section*{Standard}

Input values are not checked against any specified constraints.
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.
\begin{tabular}{ll} 
Custom & \begin{tabular}{l} 
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.
\end{tabular} \\
The previous value of the variable can be referenced by the variable name. \\
Example: \\
\(\$>=\) abct1 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.
\end{tabular}

In the Value List group it is specified if and how to present a list of values for an input box.
No List 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 A scroll control will be available to the user for the selected variable in the Scroll

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.

\section*{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.

\section*{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.}

\subsection*{30.7 Document Properties: Security}


\section*{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:

Reduce data Permission to use the command Reduce Data (page 56).

Add Sheets
Edit Script Permission to access the Edit Script Dialog (page 173) dialog.

\section*{Reload}

Permission to add more sheets to the document.

Permission to run the load script.

\section*{Note!}

The permissions to Reload and Partial Reload (see below) data are automatically disabled by the operation Reduce Data - Keep Possible Values, which is performed from the File Menu.
\begin{tabular}{|c|c|}
\hline Partial Reload & Permission to run a Partial Reload (page 56). \\
\hline Edit Module & Permission to access the Edit Module (page 875) dialog. \\
\hline Save Document (Users) & Permission to save the document with USER privileges. \\
\hline Access Document Properties (Users) & Permission to access the Settings: Document Properties with USER privileges. \\
\hline Promote/Demote Sheets & Permission to use the Promote Sheet and Demote Sheet commands. \\
\hline Allow Export & Permission to use any Export, Print or Copy to Clipboard commands. \\
\hline Allow Print (When Export Is Prohibited) & Overrules the Allow Export settings for printing purposes (only). \\
\hline Access Tabrow Properties & Permission to access the Tabrow Properties. \\
\hline Macro Override Security & Overrides any security settings by means of macros and commands via automation. \\
\hline Show All Sheets and Objects & Overrules conditional display of sheets and sheet objects, thus everything becomes visible. This functionality can be toggled by pressing CTRL+SHIFT+S. \\
\hline Show Progress for Hidden Script & Show it in the script progress dialog while executing hidden script. \\
\hline Allow User Reload & Permission to reload the script when the document is opened in USER mode. When it is deselected, reload is not possible, even if the Reload check box above is selected. \\
\hline Admin Override Security & Overrides any security settings on document or sheet level by logging in as Administrator. \\
\hline Module Password & Allowing to set a password for accessing the Edit Module (page 875) Dialog. To change the password, click the Module Password button again. \\
\hline
\end{tabular}

Further privileges for sheet objects etc. can be set in the Sheet Properties: Security (page 451) property page.

\subsection*{30.8 Document Properties: Triggers}


\section*{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 875). Not all actions can be invoked from the following event triggers. Read more about the different actions under Add Action Dialog (page 532).

\section*{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 883) before designing server documents with macro triggers.

\section*{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.

\section*{OnOpen}

Select this event to assign an action to be executed each time the QlikView document is opened. This event will not function when running in the Ajax Client.

\section*{OnPostReduceData}

Select this event to assign an action to be executed after each time the Reduce
Data command has been executed.

\section*{OnPostReload}

Select this event to assign an action to be executed each time the script has been re-executed.

\section*{Add Action(s)}

This button opens the Actions (page 532) 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 875) 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.

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 in it. The buttons open the Actions (page 532) 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 875) dialog. For more information about adding actions, see Actions (page 532).
When an action has been assigned to the field, the button changes to Edit
Action. The following Field Event Triggers exist:

\section*{OnSelect}

The action will be executed each time a selection has been made in the specified field.

\section*{OnLock}

The action will be executed each time the field is locked.

\section*{OnChange}

The action will be executed each time a selection has been made in any field which is logically associated with the specified field.

\section*{OnUnlock}

The action will be executed each time the field is unlocked.

\section*{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 532) 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 532).

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:

\section*{OnInput}

The action will be executed each time a new value is directly entered in the selected variable.

\section*{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.

\section*{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.

\section*{Note!}

Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

\subsection*{30.9 Document Properties: Groups}


\section*{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-Hierarchic Groups (Cyclic) (page 854) and Hierarchic Groups (Drill-Down) (page 853). 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.
\begin{tabular}{ll} 
New & \begin{tabular}{l} 
Clicking this button opens the Group Settings (page 428) dialog where it is \\
possible to define a new field group.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Deletes the selected group.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Opens the Group Settings (page 428) dialog for the selected group. This page \\
is used for accessing field groups.
\end{tabular}
\end{tabular}

\section*{Group Settings}


\section*{The Group Settings dialog}

This dialog is opened when clicking the New... or Edit... buttons in the Document Properties: Groups (page 427) dialog.
\begin{tabular}{ll} 
Group Name & \begin{tabular}{l} 
Enter a name for the group and select either Drill-down Group or Cyclic \\
Group.
\end{tabular} \\
Available Fields & \begin{tabular}{l} 
Available fields are listed in this column. Select the fields to include in the \\
group.
\end{tabular} \\
\begin{tabular}{l} 
Show Fields from \\
Table
\end{tabular} & \begin{tabular}{l} 
Select a table.
\end{tabular} \\
Add & \begin{tabular}{l} 
Moves the selected fields from the column of Available Fields to the column \\
of Used Fields.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Add All & \begin{tabular}{l} 
Moves all fields from the column of Available Fields to the column of Used \\
Fields.
\end{tabular} \\
Remove & \begin{tabular}{l} 
Moves the selected fields listed in the column of Used Fields to the column of \\
Available Fields.
\end{tabular} \\
Moves the selected field one step further up in the list of Used Fields. \\
Demote & \begin{tabular}{l} 
Moves the selected field one step further down in the list of Used Fields.
\end{tabular} \\
Used Fields & \begin{tabular}{l} 
Fields included in the current group are listed in this column. Initially the col- \\
umn is empty.
\end{tabular} \\
Add Expression & \begin{tabular}{l} 
Opens the Edit Expression Dialog (page 737) dialog for easy editing of long \\
formulas.
\end{tabular} \\
Edit... & \begin{tabular}{l} 
Opens the Edit Expression Dialog (page 737) dialog for editing the selected \\
formula in the Used Fields area.
\end{tabular} \\
Label & \begin{tabular}{l} 
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 For- \\
mula (page 851) for dynamic update.
\end{tabular} \\
Sort Orders... & \begin{tabular}{l} 
Clicking on this button opens the Group Sort Order dialog in which a sep- \\
arate sort order can be set for each field in the group.
\end{tabular}
\end{tabular}

\subsection*{30.10 Document Properties: Tables}
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|}
\hline \multicolumn{11}{|l|}{Document Properties [The Cars Data Base]} \\
\hline Presentation & Number & \multicolumn{2}{|r|}{Scrambly} & \multicolumn{2}{|l|}{Extensions} & Font & \multicolumn{2}{|r|}{Layout} & \multicolumn{2}{|r|}{Caption} \\
\hline General & Sheets & Server & Scheduler & Variables & Security & Triggers & \multicolumn{2}{|c|}{Groups} & Tables & Sort \\
\hline \multicolumn{11}{|l|}{Tables} \\
\hline Name & \multicolumn{2}{|r|}{Loosely Coupled} & \# Records & \multirow[t]{2}{*}{\[
\begin{aligned}
& \text { \# Fields } \\
& 9
\end{aligned}
\]} & \# Keys & \multicolumn{3}{|c|}{Comment} & & \\
\hline Test & \multicolumn{2}{|r|}{\(\square\)} & 201 & & - & & & & & \\
\hline \multicolumn{11}{|l|}{Fields} \\
\hline - Name & Dim... & Mea... & Tags & \multicolumn{2}{|l|}{Comment} & \# Tables & \# Val... & \# Disti... & \multicolumn{2}{|l|}{Type} \\
\hline 6 Class & & - & \multirow[t]{9}{*}{\begin{tabular}{l}
Sascii, Stext \\
Sascii. Stext \\
Sascii. Stext \\
Sascii, Stext
\end{tabular}} & \multicolumn{2}{|l|}{\multirow[t]{2}{*}{}} & 1 & 201 & \multicolumn{3}{|l|}{4} \\
\hline 7 Type & & - & & & & 1 & 201 & \multicolumn{3}{|l|}{12} \\
\hline 8 SubType & & - & & & & 1 & 201 & \multicolumn{3}{|l|}{5} \\
\hline 9 Id & \(\square\) & - & & & & 1 & 201 & \multicolumn{3}{|l|}{127} \\
\hline 10 Bytes & & & & & & 1 & 201 & \multicolumn{3}{|l|}{87} \\
\hline 11 Count & & \(\square\) & & & & 1 & 201 & \multicolumn{3}{|l|}{31} \\
\hline 12 Size & & & & & & 1 & 201 & \multicolumn{3}{|l|}{56} \\
\hline 13 CalcTime & & & & & & 1 & 201 & \multicolumn{3}{|l|}{3} \\
\hline 14 AvgCalcTime & & \(\square\) & & & & 1 & 201 & \multicolumn{2}{|l|}{6} & \\
\hline Edit Tags... & & & & & & & & & & \\
\hline Export Structure & Tag Clean-U & pp & & & & & & & & \\
\hline & & & & & OK & Cance & & Apply & & Help \\
\hline
\end{tabular}

\footnotetext{
Document Properties, Tables
}

\section*{Loosely Coupled Tables}

Any table can 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 253) 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 Loosely Coupled Tables (page 118).

\section*{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:
\begin{tabular}{ll} 
Name & The name of the internal table. \\
Loosely Coupled & \begin{tabular}{l} 
If the alternative is checked, the table is loosely coupled. It is possible to delib- \\
erately set this alternative for a table here.
\end{tabular} \\
\# Records & \begin{tabular}{l} 
The number of records (rows) of the table.
\end{tabular} \\
\# Fields & The number of fields (columns) of the table. \\
\# Keys & The number of key (connecting) fields (columns) of the table. \\
Comment & \begin{tabular}{l} 
Displays the comments read from the data source and the Comment Field \\
(page 226) made on the field.
\end{tabular}
\end{tabular}

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:
\begin{tabular}{ll} 
\# & \begin{tabular}{l} 
The internal number of the field. Numbers 0 to 5 are used by the QlikView sys- \\
tem fields, which are not displayed in this list.
\end{tabular} \\
Name & \begin{tabular}{l} 
The name of the field.
\end{tabular} \\
Dimensions & \begin{tabular}{l} 
Mark the check box to the right of the field name to add the system tag \$dime- \\
nsion to the field. This tag denotes a field recommended for use in chart dimen- \\
sions, list boxes etc. A field tagged with dimension will be displayed at the \\
top of all field selection controls in QlikView except in the Edit Expression \\
dialog.
\end{tabular} \\
Measures & \begin{tabular}{l} 
Mark the check box to the right of the field name to add the system tag \$mea- \\
sure to the field.. This tag denotes a field recommended for use in expressions. \\
A field tagged with measure will be displayed at the top of all field selection \\
controls in the Edit Expression dialog.
\end{tabular} \\
Tags & \begin{tabular}{l} 
Displays the Field Tags (page 169). \$ denotes a system tag.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Comment & \begin{tabular}{l} 
Displays the comments read from the data source and an indicator for any spe- \\
cial status of the field, such as Semantic, AndMode, AlwaysOneSelected, \\
Info, Locked or Hidden.
\end{tabular} \\
\# Tables & The number of tables in which the field occurs. \\
\# Values & \begin{tabular}{l} 
The total number of field values, disregarding selections. This information is \\
not available for key (connecting) fields.
\end{tabular} \\
\# Distinct & \begin{tabular}{l} 
The total number of distinct field values, disregarding selections. \\
Displays an indicator for any special status of the field, such as Semantic, And- \\
Mode, AlwaysOneSelected, Locked or Hidden. See also: System Variables \\
(page 274).
\end{tabular} \\
Edit Tags... & \begin{tabular}{l} 
Opens a dialog where it is possible to add and remove tags. System tags can- \\
not be removed here. Tags added may not be given a name occupied by a sys- \\
tem tag.
\end{tabular} \\
Export Structure & \begin{tabular}{l} 
By pressing this button it is possible to export the table structure of the doc- \\
ument to a set of text files. These text files, one for the tables (file-name.T- \\
ables.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.
\end{tabular} \\
Tag Clean-Up & \begin{tabular}{l} 
Click this button to clean up any tags that might be left after a field has been \\
removed from the QlikView document.
\end{tabular} \\
\hline
\end{tabular}

\subsection*{30.11 Document Properties: Sort}


\section*{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 493).)
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.
\begin{tabular}{ll} 
State & \begin{tabular}{l} 
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: selected, possible, excluded (ascending).
\end{tabular} \\
Expression & \begin{tabular}{l} 
Field values will be sorted according to an arbitrary expression that is entered \\
into the text edit box below this sort option.
\end{tabular} \\
Frequency & \begin{tabular}{l} 
Toggles the status whether the sorting procedure of values should be made by \\
frequency.
\end{tabular} \\
Numeric Value & \begin{tabular}{l} 
Toggles the status whether the sorting procedure of values should be made by \\
numeric value.
\end{tabular} \\
Text & \begin{tabular}{l} 
Toggles the status whether the sorting procedure of values should be made in \\
alphabetical order according to the ASCII standard.
\end{tabular} \\
Load Order & \begin{tabular}{l} 
Toggles the status whether the sorting procedure of values should be made in \\
load order.
\end{tabular}
\end{tabular}

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.

\subsection*{30.12 Document Properties: Presentation}


\section*{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.

\section*{Default Theme for New Objects}

\section*{Alignment}

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 QlikView 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.

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.
\(\left.\begin{array}{ll}\text { Multiline Cells } & \begin{array}{l}\text { If the Wrap Text option is checked, the (text) contents of the cell (field value) } \\ \text { will be displayed in two or more lines. } \\ \text { The number of cell lines is set in the Cell Height ( Lines ) edit box. }\end{array} \\ \text { Settings } & \begin{array}{l}\text { Single Column } \\ \text { Checking this option forces single column mode. } \\ \text { Order by Column } \\ \text { In multiple column displays the field values will be ordered by column, rather } \\ \text { than by row. } \\ \text { Cell Borders }\end{array} \\ \text { Checking this option will introduce horizontal lines between field values. }\end{array}\right\}\)

\subsection*{30.13 Document Properties: Number}


\section*{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:
\begin{tabular}{ll} 
Mixed & Both numbers and text. Numbers shown in original format. \\
Number & \begin{tabular}{l} 
Shows numeric values with the number of digits set in the Precision spinner \\
box.
\end{tabular} \\
Integer & \begin{tabular}{l} 
Shows numeric values as integers.
\end{tabular} \\
Fixed to & \begin{tabular}{l} 
Shows numeric values as decimal values with the number of decimal digits set \\
in the Decimals spinner box.
\end{tabular} \\
Money & \begin{tabular}{l} 
Shows numeric values in the format shown in the Preview text box. The \\
default format is the Windows Currency setting.
\end{tabular} \\
Date & \begin{tabular}{l} 
Shows values that can be interpreted as dates in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Time & \begin{tabular}{l} 
Shows values that can be interpreted as time in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Timestamp & \begin{tabular}{l} 
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.
\end{tabular} \\
Interval & \begin{tabular}{l} 
Shows time as sequential time increment (e.g. format = mm shows the value as \\
the number of minutes since calendar start ( \(1899: 12: 30: 24: 00)\).
\end{tabular}
\end{tabular}

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.

\subsection*{30.14 Document Properties: Scrambling}


\section*{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.

Fields

\section*{Scramble}

This is a list of all fields in the document. Select one or more fields for scrambling.

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.

\subsection*{30.15 Font}


\section*{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 361). 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.

\subsection*{30.16 Layout}


\section*{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.

\section*{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.
\begin{tabular}{ll} 
Shadow Intensity & \begin{tabular}{l} 
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.
\end{tabular} \\
Border Style & \begin{tabular}{l} 
The following predefined border types are available: \\
Solid \\
A solid unicolored border. \\
Depressed \\
Border giving the impression of depressing the sheet object from the back- \\
ground. \\
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.
\end{tabular} \\
Change Image... & \begin{tabular}{l} 
If the Image border type is selected, click this button to set the image you \\
wish to use.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Stretch Image & \begin{tabular}{l} 
If the Image border type is used and this check box is marked, the image pat- \\
tern 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.
\end{tabular} \\
Border Width & \begin{tabular}{l} 
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).
\end{tabular} \\
Color & \begin{tabular}{l} 
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.
\end{tabular} \\
Rainbow & \begin{tabular}{l} 
Creates a rainbow colored border for all border types except Image. The rain- \\
bow will start with the selected base color on top of the sheet object.
\end{tabular}
\end{tabular}

When Simplified is the chosen in Document Properties: General (page 408), there is no choice of border type, there is only the Shadow Intensity drop-down menu and the Border Width setting.

\section*{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:
\begin{tabular}{ll} 
Rounded Corners & \begin{tabular}{l} 
Checking this option makes the alternatives for rounded corner shapes pos- \\
sible.
\end{tabular} \\
Corners & \begin{tabular}{l} 
Corners for which the check box remains unmarked will be drawn rectangular \\
instead.
\end{tabular} \\
Squareness & \begin{tabular}{l} 
A variable number between 2 and 100 where 100 defines a rectangle with per- \\
fectly 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.
\end{tabular} \\
Corner Radius & \begin{tabular}{l} 
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).
\end{tabular}
\end{tabular}

\section*{Layer}

In the Layer group an object can be defined as residing in one of three layers:
\begin{tabular}{ll} 
Bottom & \begin{tabular}{l} 
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.
\end{tabular} \\
Normal & \begin{tabular}{l} 
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 Bot- \\
tom layer and can never obscure sheet objects in the Top layer.
\end{tabular} \\
Top & \begin{tabular}{l} 
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.
\end{tabular} \\
Custom & \begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

\section*{Show}

In the Show group it is possible to specify a condition under which the sheet object is displayed:
\begin{tabular}{ll} 
Always & The sheet object will always be displayed. \\
Conditional & \begin{tabular}{l} 
The sheet object will be shown or hidden depending on a Conditional Func- \\
tions (page 328) which will be evaluated continuously depending on e.g. \\
selections etc. The sheet object will only be visible when the condition returns
\end{tabular} \\
TRUE.
\end{tabular}

\section*{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 423). This functionality can be toggled by pressing CTRL+SHIFT+S.

\section*{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.
\begin{tabular}{ll} 
Allow Move/Size & \begin{tabular}{l} 
If this option has been deselected it will be impossible to move or resize the \\
sheet object.
\end{tabular} \\
Allow Copy/Clone & \begin{tabular}{l} 
If this option has been deselected it will be impossible to make a copy of the \\
sheet object.
\end{tabular} \\
Allow Info & \begin{tabular}{l} 
When the Info (page 239) 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.
\end{tabular} \\
Size to Data & \begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

\section*{Scrollbars}

Various controls for changing the scrollbar layout are located in the Scrollbars group:
Preserve Scroll Posi- With this setting enabled, QlikView will try to preserve the scroll position of tion

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 412) dialog that opens when you click the appropriate button.

Scroll Background Sets the scroll bar background color. Select a color by clicking the button. Note that medium gray tones often render the best results for scroll bars. Either color can be defined as a solid color or a gradient via the Color Area (page 412) 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.
\begin{tabular}{ll} 
Scroll Style & \begin{tabular}{l} 
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.
\end{tabular} \\
Apply To... & \begin{tabular}{l} 
Opens the Caption and Border Properties dialog where you can set where to \\
apply the properties that have been set on the Layout page.
\end{tabular}
\end{tabular}

\section*{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 477).
3. The area in the middle has no function and may contain anything or nothing.


\section*{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.

\section*{Caption and Border Properties}


\section*{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:

\section*{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.

\section*{Objects in this document}

Applies the settings to objects of the entire document.
2. Choose one of the options:

\section*{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.

\section*{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.

\subsection*{30.17 Caption}


\section*{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.

\section*{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 412) 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 361).

\section*{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:
\begin{tabular}{ll} 
X-pos & \begin{tabular}{l} 
Sets the horizontal position of the left side of sheet object in relation to the \\
sheet's left edge.
\end{tabular} \\
Y-pos & \begin{tabular}{l} 
Sets the vertical position of the top side of sheet object in relation to the \\
sheet's upper edge.
\end{tabular} \\
Width & Sets the width of the QlikView sheet object. \\
Height & Sets the height of the QlikView sheet object.
\end{tabular}

The orientation of the caption label can be altered with the Caption Alignment options:
Horizontal The label can be horizontally aligned: Left, Centered or Right within the caption area.

Vertical The label can be vertically aligned: Top, Centered or Bottom within the caption area.

\section*{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.

\section*{Note!}

Use the special caption icons with care. Too many icons will only confuse the user.
\begin{tabular}{ll} 
Allow Minimize & \begin{tabular}{l} 
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-click- \\
ing the caption.
\end{tabular} \\
Auto Minimize & \begin{tabular}{l} 
This option becomes available when Allow Minimize is checked. When Auto \\
Minimize 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.
\end{tabular}
\end{tabular}

\section*{Allow Maximize}

Help Text

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 doubleclicking the caption. If both Allow Minimize and Allow Maximize are checked, double-clicking will effect minimizing of the object.

Here you may enter a help text to be displayed in a pop-up window. The help text may be specified as a Calculated Formula (page 851). This option is not available at document level. Click the ... button to open the Edit Expression 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.

\section*{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.

\subsection*{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.

\subsection*{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:
\begin{tabular}{ll} 
Tabrow Properties... & \begin{tabular}{l} 
Opens a dialog in which you can choose a font for the tab. Only Truetype \\
fonts may be used. See Tabrow Properties (page 453).
\end{tabular} \\
Sheet Properties... & \begin{tabular}{l} 
Selects the sheet and opens the Sheet Properties dialog. Here sheet specific \\
parameters such as the sheet background, the fields to display, the default font \\
and the object look etc. can be specified. This alternative is only available \\
when clicking on the tab belonging to the currently active sheet.
\end{tabular} \\
Copy Sheet & \begin{tabular}{l} 
Makes a complete copy of the entire sheet with all its sheet objects. The copy \\
will be named "Copy of Sheetname" and positioned as the last sheet in the \\
document.
\end{tabular} \\
Promote Sheet & \begin{tabular}{l} 
Moves the sheet one step further to the left.
\end{tabular} \\
Demote Sheet & \begin{tabular}{l} 
Moves the sheet one step further to the right.
\end{tabular} \\
Help & Opens context-specific help. \\
Remove & Makes the sheet active, then removes it.
\end{tabular}

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.

\subsection*{31.3 Sheet: Object Menu}
\begin{tabular}{ll} 
Properties & \begin{tabular}{l} 
Opens the Sheet Properties dialog page where you can set the parameters \\
defining the sheet.
\end{tabular} \\
Select Fields & \begin{tabular}{l} 
Opens the Sheet Properties: Fields page where you can select a field or more \\
to be displayed as a list box on the sheet.
\end{tabular} \\
New Sheet Object & \begin{tabular}{l} 
Opens a cascading menu listing sheet objects. \\
Makes a complete copy of the entire sheet with all its sheet objects. The copy \\
will be named "Copy of Sheetname" and positioned as the last sheet in the \\
document.
\end{tabular} \\
Copy Sheet & \begin{tabular}{l} 
Pastes a sheet object, previously copied to clipboard, on the sheet. This com- \\
mand can also be invoked by the following keyboard shortcut: CTRL+V.
\end{tabular} \\
Paste Sheet Object Sheet Object as & \begin{tabular}{l} 
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.
\end{tabular} \\
Link & \begin{tabular}{l} 
Print will result in the printout of an image of the sheet area including any \\
sheet objects.
\end{tabular} \\
Print & \begin{tabular}{l} 
Copies a bitmap picture of the sheet area (only) to Clipboard.
\end{tabular} \\
board Image to Clip- & \begin{tabular}{l} 
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.
\end{tabular} \\
Help Image to File & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove
\end{tabular}

\subsection*{31.4 Sheet Properties}

Opens the Sheet Properties: General (page 447) 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 451)).

\subsection*{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.
\begin{tabular}{|c|c|}
\hline Alternate State & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Sheet ID & This is used for Internal Macro Interpreter (page 873) purposes. Every sheet is assigned a unique ID, starting with SH 01 . It is possible to edit this ID number later on. \\
\hline
\end{tabular}

In the Show Sheet group it is possible to specify a condition under which the sheet is displayed:
\begin{tabular}{ll} 
Always & The sheet will always be displayed. \\
Conditional & \begin{tabular}{l} 
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 423) page. \\
This functionality can be toggled by pressing CTRL+SHIFT+S.
\end{tabular}
\end{tabular}

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.

\section*{Image Formatting:}
\begin{tabular}{ll} 
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 & \begin{tabular}{l} 
The image is stretched as far as possible while retaining the correct aspect \\
ratio.
\end{tabular} \\
Fill with Aspect & \begin{tabular}{l} 
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.
\end{tabular} \\
Tiled & \begin{tabular}{l} 
If this option is selected, the image will be tiled the as many times as space \\
allows.
\end{tabular}
\end{tabular}

Horizontal and Vertical orientation:

Horizontal
Vertical The image can be vertically aligned: Top, Centered or Bottom.
Theme Maker... Opens the Theme Maker Wizard (page 583) for creating a new theme or editing an existing one. See Layout Themes (page 581) 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 581) for more details.

In the Tab Settings group you may define a background color for the sheet tab.
\begin{tabular}{ll} 
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 & \begin{tabular}{l} 
This setting lets you specify a custom Tab Color and Tab Text Color by \\
clicking the corresponding buttons.
\end{tabular} \\
Apply to All & Applies the tab color settings to all sheets of the document.
\end{tabular}

\subsection*{31.6 Select Fields/Sheet Properties: Fields}


\section*{Sheet Properties, Fields}

This is where you select Fields (page 169) 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 256) fields in the load script.) It is also possible to view the fields of one table at a time.

\subsection*{31.7 Sheet Properties: Objects}


\section*{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:

\section*{Object ID}

Type
Caption
ShowMode

\section*{State}

\section*{CalcTime}

Lists the unique Object ID of every existing sheet object. Linked objects have the same object ID.

Clarifies the object type.
Gives the text, if any, that is displayed in the caption of the object.
Indicates whether the object is normal, minimized, maximized, hidden or has a show condition.

Displays which state the object is in. See Alternate States (page 122) for additional information.

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).
\(\left.\begin{array}{ll}\text { Layer } & \begin{array}{l}\text { The layers are set on the Sheet Object Properties: Layout page where Bot- } \\
\text { tom, Normal and Top correspond to the numbers }-1,0 \text { and } 1 . \text { Choose Custom } \\
\text { to enter a value of your choice. Valid layer numbers are }-128 \text { to } 127 .\end{array} \\
\text { Memory } & \begin{array}{l}\text { The amount of transient memory needed for the last recalculation of the } \\
\text { object's contents. This information is only relevant for objects that are actually } \\
\text { calculated (charts, table boxes and any objects containing formulas). }\end{array} \\
\begin{array}{ll}\text { Left, Top, Right, } \\
\text { Bottom }\end{array} & \begin{array}{l}\text { The current position of the indicated edge of the sheet object relative to the } \\
\text { edge of the document window, expressed in pixels. }\end{array} \\
\text { Width, Height } & \text { The size dimensions of the object, expressed in pixels. }\end{array}\right\}\)\begin{tabular}{ll} 
Left (Min), Top (Min), & The current position of the object when iconized (=minimized). \\
Right (Min), \\
Bottom (Min) & The dimensions of the object's icon. \\
Width (Min), Height \\
(Min)
\end{tabular}

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.

\subsection*{31.8 Sheet Properties: Security}


\section*{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:

\footnotetext{
Add Sheet Objects Permission to add new sheet objects.
}

Remove Sheet Objects Permission to remove any of the sheet objects.
\begin{tabular}{ll}
\begin{tabular}{l} 
Move/Size Sheet \\
Objects
\end{tabular} & Permission to change the layout of existing sheet objects. \\
\begin{tabular}{l} 
Copy/Clone Sheet \\
Objects
\end{tabular} & Permission to add a copy of existing sheet objects. \\
\begin{tabular}{l} 
Access Sheet Object \\
Properties
\end{tabular} & Permission to access the Sheet Properties dialog. \\
Remove Sheet & Permission to delete the sheet. \\
\begin{tabular}{l} 
Access Sheet Properties \\
(Users)
\end{tabular} & Permission to access the different property pages of sheet objects.
\end{tabular}

Checking the Apply to All Sheets option, applies the above options to all sheets of the document.

\section*{Read Only 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 423) property page.

\subsection*{31.9 Sheet Properties: Triggers}


\section*{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 873).
Not all actions can be invoked from the following event triggers. Read more about the different actions under Add Action Dialog (page 532).

\section*{Note!}

There are limitations as to the behavior of certain macro triggers when working with documents on

QlikView Server. Please study the special section Using Macros in QV Documents on the QV-Server in the QlikView Reference Manual before designing server documents with macro triggers.

\section*{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 532) 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.

\subsection*{31.10 Tabrow Properties}

The properties of the tabrow are set in the Tabrow Properties dialog, which is opened by right-clicking the tabrow.

\footnotetext{
Font See Font (page 475). 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.
}

\section*{32 Sheet Objects}

The following types of sheet objects exist:
\begin{tabular}{ll} 
List Box & page 457 \\
Statistics Box & page 483 \\
Multi Box & page 489 \\
Table Box & page 501 \\
Charts & page 587 \\
Input Box & page 519 \\
Current Selections Box & page 513 \\
Button & page 529 \\
Text Object & page 539 \\
Line/Arrow Object & page 545 \\
Slider/Calendar Object & page 549 \\
Custom Object & page 573 \\
Search Object & page 561 \\
Bookmark Object & page 557 \\
Container & page 567
\end{tabular}

\subsection*{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 My Server Objects (page 579) page of the Server Objects Dialog (page 578) dialog, accessible from the Layout menu, or through the Server Objects Pane (page 577) pane, accessible from the View menu.
Shared server objects are only available when working with a document on QlikView Server and only to authenticated users. Any user who has created a personal server object may flag this as shared. They will then become available to other users. Just like personal server objects shared server sheet objects are stored in a repository on the server. Available shared server objects can be managed from the Shared Server Objects page of the Server Objects Dialog (page 578), accessible from the Layout menu, or through the Server Objects Pane (page 577), accessible from the View menu.

\subsection*{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.

\subsection*{32.3 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.

\section*{Note!}

The QlikView IE plug-in must be installed on the client computer where the Microsoft Office document is used.

\section*{33 List Box}
\begin{tabular}{|lr|}
\hline Country \\
Argentina & 30 \\
Austria & 2 \\
Belgium & 2 \\
Brazil & 9 \\
Canada & 3 \\
Denrark & 2 \\
Finla & 2 \\
France & \(11 \frac{1}{-}\) \\
\hline
\end{tabular}

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 457). It can also be accessed from the Object menu when the list box is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the List Box Properties dialog where List Box Properties: General \\
(page 460), List Box Properties: Expressions (page 463), List Box Properties: \\
Sort (page 493), List Box Properties: Presentation (page 471), Properties: \\
Number (page 473), Font (page 475), Layout (page 476), and Caption (page \\
479) parameters can be set.
\end{tabular} \\
Cycle & \begin{tabular}{l} 
Only available when the list box is set to display a Non-Hierarchic Groups \\
(Cyclic) (page 854). Opens a cascade menu with the fields of the group for \\
direct selection of which field to be displayed in the list box.
\end{tabular} \\
Search & \begin{tabular}{l} 
Opens the Search (page 113) box.
\end{tabular} \\
Fuzzy Search & \begin{tabular}{l} 
Opens the text search box in Fuzzy Search (page 114) mode.
\end{tabular} \\
Advanced Search & \begin{tabular}{l} 
Opens the Advanced Search Dialog (page 116). The dialog makes it possible \\
to enter advanced search expressions. This command can also be invoked by \\
the following keyboard shortcut: SHIFT+CTRL+F.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of View \\
Menu (page 58) is activated or when the Always Show Design Menu Items \\
(page 86) 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
\end{tabular} \\
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
\end{tabular}
\begin{tabular}{|c|c|}
\hline Print Possible... & Opens the Print: General (page 99) dialog where print settings can be specified. All non-excluded values of the active list box are sent to the printer. \\
\hline Print Possible as PDF... & \begin{tabular}{l}
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 homepage and installed separately.
\end{tabular} \\
\hline Send to Excel & Exports possible (selected and optional) values to Microsoft Excel, which is automatically launched if not already running. 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. \\
\hline & \begin{tabular}{l}
Note! \\
Mini charts will not be displayed when exporting to Excel!
\end{tabular} \\
\hline Export... & Opens the Save as dialog where path, file name and (table) file type for the exported data content can be specified. \\
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Possible Values \\
Copies the possible (selected and optional) values to the clipboard. \\
Cell Value \\
Copies the text value of the list box cell right-clicked upon (when invoking the Object menu) to the clipboard. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the User Preferences: 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\square\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline
\end{tabular}

\subsection*{33.2 List Box Properties: General}


\section*{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.

\section*{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 851) 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 427) dialog and will be listed with the available fields.
If a Hierarchic Groups (Drill-Down) (page 853) 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 854) 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 602). 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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Note!}

The use of calculated fields in list boxes may cause performance issues with very large data sets.
\begin{tabular}{ll} 
And mode & A Multiple Selection Within a Field (page 109) 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 117). This option is not available for calculated fields.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Show Alternatives & Toggles whether a selection directly in the active field list box is to exclude the other values of the field or not. If Show Alternatives 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. \\
\hline Hide Excluded & Toggles whether the excluded values of the field should be shown or not. Excluded values become unselectable as well. \\
\hline Read Only & Checking this option will prevent you from making selections in the list box. However, it will still reflect selections made elsewhere in the document. \\
\hline Always One Selected Value & Checking this option assures that one value will always 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. \\
\hline Override Locked Field & 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. \\
\hline Show as TreeView & Shows the list box as a List Box Tree View (page 481). This control is only relevant if the field contains the path representation of nodes in a hierarchy. Such a field can be generated using the Path parameter of the Hierarchy prefix. \\
\hline With Separator & Sets the character that should be interpreted as separator in the path used for the Tree View. \\
\hline Show Frequency & 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. \\
\hline In Percent & Toggles the status for whether the frequency should be shown in absolute numbers or as percentages of the total number of entries. \\
\hline Alternate State & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Object ID & This is used for Internal Macro Interpreter (page 873) 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. \\
\hline Include Excluded Values in Search & The drop-down list lets you choose between Yes, No or <use default> which applies the default setting as specified under User Preferences: General. \\
\hline
\end{tabular}

\section*{Default Search Mode}

\section*{Print Settings}

\section*{Print All Values}

Layout Dependent Print

Specifies the initial default search mode to be used in text searches. The mode can always be changed on the fly by typing * or \(\sim\) as part of the search string. The following alternatives are available:

\section*{Use <default>}

The default as specified under User Preferences (page 77) will apply.
Use Wildcard Search
The initial search string will be two wildcards with the cursor between them to facilitate a wildcard search.

\section*{Use Fuzzy Search}

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

\section*{Use Normal Search}

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

\section*{Use Associative Search}

Associative search will be used in the search box. For more information about associative search, see Search (page 113).

Here you set the print settings for the print job.
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.

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.

\subsection*{33.3 List Box Properties: Expressions}


\footnotetext{
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.

\section*{The Expression List}

\section*{Navigating the Tree Controls:}

The expression list is in fact a tree control with a large set of control options. In front of each expression you will find an expansion icon. By clicking on the expansion icon you open up underlying subexpressions or attribute expressions. The icon will then shift to a collapse icon. By clicking that icon you collapse the display of any sub expressions or attribute expressions. In complex cases up to three levels of expansion may be possible.

\section*{Attribute Expressions:}

Expression data can be dynamically formatted by means of attribute expressions. Click on the expansion icon in front of any expression to display the placeholders for the dimension's attribute expressions. You may choose to use any number and combination of attribute expressions on a given base expression. Whenever an attribute expression is entered for a dimension, its icon will turn from gray scale to color, or as in the case of Text Format, from gray to black. Formatting defined by means of attribute expression supersede default formatting of the data plots in terms of color etc.

Click here to read more about the available attribute expression types.
\begin{tabular}{ll} 
Background color & \begin{tabular}{l} 
Click on Background Color in order to define an attribute expression for cal- \\
culating 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 Color Functions (page 361). If the result of the \\
expression is not a valid color representation, black will be used.
\end{tabular} \\
Text Color & \begin{tabular}{l} 
Click on Text Color in order to define an attribute expression for calculating \\
the text color of text associated with the data point. The expression used \\
should return a valid color representation (a number representing the Red, \\
Green and Blue components as defined in Visual Basic), which is typically \\
achieved by using one of the special Color Functions (page 361). If the result \\
of the expression is not a valid color representation, black will be used.
\end{tabular} \\
Text Format & \begin{tabular}{l} 
Click on Text Format in order to enter an attribute expression for calculating \\
the font style for text associated with the data point. The expression used as \\
text format expression should return a string containing a '<B>' for bold text, \\
'<I>' for Italic text and/or '<U>' for underlined text.
\end{tabular} \\
Pie Popout & \begin{tabular}{l} 
Only applicable on pie charts. Click on Pie Popout in order to enter an attrib- \\
ute expression for calculating whether the pie slice associated with the data
\end{tabular} \\
point should be drawn in an extracted "popout" position.
\end{tabular}
Line Style
\begin{tabular}{l} 
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 \(<\mathrm{W} n>\) 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: \(<\mathrm{W} 2.5>\). The style of the line can be con- \\
trolled by including a tag \(<\mathrm{S} n>\) where \(n\) is an integer between 1 and 4 indi- \\
cating the style to be used \((1=\) continuous, \(2=\) dashed, \(3=\) dotted, \\
\(4=\) dashed/dotted). Example: \(<\) S3 \(>\). The \(<\mathrm{W} n>\) and \(<\) Sn \(>\) tags can be freely com- \\
bined, but only the first occurrence of each counts. The tags must be enclosed \\
by single quotations.
\end{tabular}
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

Delete

\section*{Export...}

\section*{Paste}

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 ute 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.
By clicking this button, you get to the Edit Expression Dialog (page 737) 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.

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.

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.

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.
\(\left.\begin{array}{ll}\text { Import... } & \begin{array}{l}\text { The option is only available in the context menu that appears when right-click- } \\
\text { ing in the blank area in the list of expressions. The command opens a dialog } \\
\text { where you can browse to previously exported expressions. The imported } \\
\text { expression will appear as a new main expression in the chart. }\end{array} \\
\text { Promotes the selected expression one step up in the list. Use this button if you } \\
\text { want to change the order of the expressions. } \\
\text { Demotes the selected expression one step down in the list. Use this button if } \\
\text { you want to change the order of the expressions. }\end{array}\right\}\)\begin{tabular}{l} 
The symbol entered here will be used for displaying NULL values in the table. \\
Dull Symbol \\
Missing Symbol \\
The symbol entered here will be used for displaying missing values in the \\
table. \\
Shows the composition of the selected expression. It is possible to edit the \\
expression directly in this box. By clicking the ... button the full Edit Expres- \\
sion Dialog (page 737) dialog is opened.
\end{tabular}

\section*{Representation}

Gauge Settings..

\section*{Set Default Colors}

Force Zero Based Scaling

The following alternatives are given:
Text
When selecting this option the expression values will always be interpreted and displayed as text.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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).

\section*{Note!}

The mini chart will not be displayed when exporting to Excel.
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 702) 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 468).

Sets the colors for the highlight settings to QlikView default colors.
Fixes the lower edge of the chart to zero on the axis. Not available for whiskers.

Force Shared Scale for Forces all cells in the column to use the same y-axis scale.
Y-axis

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Appearance}
\begin{tabular}{ll} 
Mode & Set the mini chart as Sparklines, line with Dots, Dots, Bars or Whiskers. \\
Color & \begin{tabular}{l} 
Opens the Color Area (page 412) dialog where the color of the mini chart can \\
be set.
\end{tabular} \\
\begin{tabular}{l} 
Highlight Max Value \\
With Color
\end{tabular} & \begin{tabular}{l} 
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.
\end{tabular} \\
\begin{tabular}{l} 
Highlight Min Value \\
With Color
\end{tabular} & \begin{tabular}{l} 
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.
\end{tabular} \\
Highlight Start Values & \begin{tabular}{l} 
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.
\end{tabular} \\
With Color & \begin{tabular}{l} 
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.
\end{tabular} \\
Highlight End Values \\
With Color & \begin{tabular}{l} 
Sets the colors for the highlight settings to QlikView default colors.
\end{tabular} \\
Set Default Colors & \begin{tabular}{l} 
Fixes the lower edge of the chart to zero on the axis. Not available for Whisk- \\
ers.
\end{tabular} \\
\begin{tabular}{ll} 
Force Zero Based Scal \\
ing
\end{tabular} \\
Force Shared Scale for & \begin{tabular}{l} 
Forces all cells in the column to use the same y-axis scale.
\end{tabular} \\
Y-Axis
\end{tabular}

\subsection*{33.4 List Box Properties: Sort}


\section*{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:
\begin{tabular}{ll} 
State & \begin{tabular}{l} 
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.
\end{tabular} \\
Expression & \begin{tabular}{l} 
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. See also Set Analysis (page 799).
\end{tabular} \\
Frequency & \begin{tabular}{l} 
Sorts the field values by frequency (number of occurrences in the table).
\end{tabular} \\
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.
\end{tabular}

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.

\subsection*{33.5 List Box Properties: Presentation}


\section*{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: \(\backslash \mathrm{Mypic} . \mathrm{jpg}\) ) 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.

\section*{No Stretch}

Fill
Keep Aspect

Fill with Aspect

The image is displayed as is, which may cause masking problems.
The image is stretched in order to fill the cell, without regard to aspect ratio.
The image is stretched as far as possible while retaining the correct aspect ratio.

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.

\section*{Single Column}

Suppress Horizontal Scroll Bar

Fixed Number of Columns

Order by Column

\section*{Cell Borders}

Highlight Search Substring

Background...

\section*{Alignment}

\section*{Multiline Settings}

With this option checked, the list box field values will always be presented in a single column.

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 list box. Instead, field values will be truncated as needed.

Check this option to set a fixed number of columns for the list box. The option is unavailable when Single Column is selected.

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.

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.

The search string entered in a search object so far will be highlighted in the matches.

Opens the Background Settings dialog.
Here you set the alignment of field values in the list box. The alignment of Text and Numbers is set separately.

In this group the list box cells may be set to display values in multiple rows, which is useful for long text strings.

\section*{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.

\section*{Background Settings}


The Background Settings dialog
\(\left.\begin{array}{ll}\text { Color } & \begin{array}{l}\text { If this setting is enabled the background of the object will be colored. Choose } \\
\text { a color by clicking on the Color button. }\end{array} \\
\text { Color Transparency } & \begin{array}{l}\text { Only available when the color option has been enabled. Sets the transparency } \\
\text { of the background color. }\end{array} \\
\text { Image } & \text { If this setting is enabled, the background of the object will be an image. Click } \\
\text { Browse to choose an image. Click Remove to remove it from the background. }\end{array}\right\}\)\begin{tabular}{l} 
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
\end{tabular}

\subsection*{33.6 Properties: Number}


\footnotetext{
Change Document Format.
}

\section*{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 434) 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:

Mixed
Number Shows numeric values with the number of digits set in the Precision spinner box.

\section*{Integer Shows numeric values as integers.}

Fixed to Shows numeric values as decimal values with the number of decimal digits set in the Decimals spinner box.

Money Shows values in the format set in the Format Pattern edit box. An example of this format is shown in the Preview text box. The default format is the Windows Currency setting.

Date Shows values that can be interpreted as dates in the format set in the Format Pattern edit box. An example of this format is shown in the Preview text box.

Time \(\quad\) Shows values that can be interpreted as time in the format set in the Format Pattern edit box. An example of this format is shown in the Preview 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 \(=m m\) shows the value as the number of minutes since calendar start (1899:12:30:24:00).

Show in Percent (\%) This check box affects the following formats: Number, Integer and Fixed.
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 434) page where the default number format of any field can be edited.

\subsection*{33.7 Font}


\section*{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 361). 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.

\subsection*{33.8 Layout}


\section*{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.

\section*{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.
\begin{tabular}{ll} 
Shadow Intensity & \begin{tabular}{l} 
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.
\end{tabular} \\
Border Style & \begin{tabular}{l} 
The following predefined border types are available: \\
Solid \\
A solid unicolored border. \\
Depressed \\
Border giving the impression of depressing the sheet object from the back- \\
ground. \\
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.
\end{tabular} \\
Change Image... & \begin{tabular}{l} 
If the Image border type is selected, click this button to set the image you \\
wish to use.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Stretch Image & \begin{tabular}{l} 
If the Image border type is used and this check box is marked, the image pat- \\
tern 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.
\end{tabular} \\
Border Width & \begin{tabular}{l} 
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).
\end{tabular} \\
Color & \begin{tabular}{l} 
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.
\end{tabular} \\
Rainbow & \begin{tabular}{l} 
Creates a rainbow colored border for all border types except Image. The rain- \\
bow will start with the selected base color on top of the sheet object.
\end{tabular}
\end{tabular}

When Simplified is the chosen in Document Properties: General (page 408), there is no choice of border type, there is only the Shadow Intensity drop-down menu and the Border Width setting.

\section*{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:
\begin{tabular}{ll} 
Rounded Corners & \begin{tabular}{l} 
Checking this option makes the alternatives for rounded corner shapes pos- \\
sible.
\end{tabular} \\
Corners & \begin{tabular}{l} 
Corners for which the check box remains unmarked will be drawn rectangular \\
instead.
\end{tabular} \\
Squareness & \begin{tabular}{l} 
A variable number between 2 and 100 where 100 defines a rectangle with per- \\
fectly 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.
\end{tabular} \\
Corner Radius & \begin{tabular}{l} 
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).
\end{tabular}
\end{tabular}

\section*{Layer}

In the Layer group an object can be defined as residing in one of three layers:
\begin{tabular}{ll} 
Bottom & \begin{tabular}{l} 
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.
\end{tabular} \\
Normal & \begin{tabular}{l} 
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 Bot- \\
tom layer and can never obscure sheet objects in the Top layer.
\end{tabular} \\
Top & \begin{tabular}{l} 
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.
\end{tabular} \\
Custom & \begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

\section*{Show}

In the Show group it is possible to specify a condition under which the sheet object is displayed:
\begin{tabular}{ll} 
Always & The sheet object will always be displayed. \\
Conditional & \begin{tabular}{l} 
The sheet object will be shown or hidden depending on a Conditional Func- \\
tions (page 328) which will be evaluated continuously depending on e.g. \\
selections etc. The sheet object will only be visible when the condition returns
\end{tabular} \\
TRUE.
\end{tabular}

\section*{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 423). This functionality can be toggled by pressing CTRL+SHIFT+S.

\section*{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.
\begin{tabular}{ll} 
Allow Move/Size & \begin{tabular}{l} 
If this option has been deselected it will be impossible to move or resize the \\
sheet object.
\end{tabular} \\
Allow Copy/Clone & \begin{tabular}{l} 
If this option has been deselected it will be impossible to make a copy of the \\
sheet object.
\end{tabular} \\
Allow Info & \begin{tabular}{l} 
When the Info (page 239) 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.
\end{tabular} \\
Size to Data & \begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

\section*{Scrollbars}

Various controls for changing the scrollbar layout are located in the Scrollbars group:
Preserve Scroll Posi- With this setting enabled, QlikView will try to preserve the scroll position of tion

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 412) dialog that opens when you click the appropriate button.

Scroll Background Sets the scroll bar background color. Select a color by clicking the button. Note that medium gray tones often render the best results for scroll bars. Either color can be defined as a solid color or a gradient via the Color Area (page 412) 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.
\begin{tabular}{ll} 
Scroll Style & \begin{tabular}{l} 
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.
\end{tabular} \\
Apply To... & \begin{tabular}{l} 
Opens the Caption and Border Properties dialog where you can set where to \\
apply the properties that have been set on the Layout page.
\end{tabular}
\end{tabular}

\subsection*{33.9 Caption}


\section*{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.

\section*{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 412) 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 361).
\[
\begin{array}{ll}
\text { Wrap Text } & \text { If this option is checked, the caption will be displayed in two or more rows. } \\
\text { Caption Height (Lines) } & \text { Set the number of caption lines in this edit box. }
\end{array}
\]

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:
\begin{tabular}{ll} 
X-pos & \begin{tabular}{l} 
Sets the horizontal position of the left side of sheet object in relation to the \\
sheet's left edge.
\end{tabular} \\
Y-pos & \begin{tabular}{l} 
Sets the vertical position of the top side of sheet object in relation to the \\
sheet's upper edge.
\end{tabular} \\
Width & \begin{tabular}{l} 
Sets the width of the QlikView sheet object.
\end{tabular} \\
Height & Sets the height of the QlikView sheet object.
\end{tabular}

The orientation of the caption label can be altered with the Caption Alignment options:
Horizontal The label can be horizontally aligned: Left, Centered or Right within the caption area.

Vertical The label can be vertically aligned: Top, Centered or Bottom within the caption area.

\section*{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.

\section*{Note!}

Use the special caption icons with care. Too many icons will only confuse the user.

\section*{Allow Minimize}

\section*{Auto Minimize}

\section*{Allow Maximize}

Help Text

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.

This option becomes available when Allow Minimize is checked. When Auto Minimize 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.

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 doubleclicking the caption. If both Allow Minimize and Allow Maximize are checked, double-clicking will effect minimizing of the object.

Here you may enter a help text to be displayed in a pop-up window. The help text may be specified as a Calculated Formula (page 851). This option is not available at document level. Click the ... button to open the Edit Expression 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.

\section*{Caption and Border Properties}
\begin{tabular}{|l|l|}
\hline Caption and Border Properties \\
\(\nabla\) Apply properties to... \\
O Objects on this sheet \\
Objects in this document \\
O Only this object type \\
All object types \\
Set as default for new objects in this document
\end{tabular}

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:

\section*{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.

\section*{Objects in this document}

Applies the settings to objects of the entire document.
2. Choose one of the options:

\section*{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. objects in this document

Set as default for new 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.

\subsection*{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.

\section*{34 Statistics Box}
\begin{tabular}{|lr|}
\hline NetSales & \\
\hline Sum & \(\$ 1,565,525\) \\
Average & \(\$ 721\) \\
Min & \(\$ 2\) \\
Max & \(\$ 32,075\) \\
\hline
\end{tabular}

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 483). It can also be accessed from the Object menu, when the statistics box is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Statistics Box Properties: General (page 485) page. Here it is pos- \\
sible to choose the statistics to be presented in the statistics box. \\
The Statistics Box Properties: Number (page 487), Font (page 475), Layout \\
(page 476), and Caption and Border Properties (page 481) pages can also be \\
reached from this command. Here the font and border parameters can be set.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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
\end{tabular} \\
\begin{tabular}{l} 
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.
\end{tabular} \\
Send Backward \\
Decreases the layout layer of the sheet object by one. Minimum value is -128.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Lock & Locks the selected value（s）in the corresponding list box． \\
\hline Unlock & Unlocks the locked value（s）in the corresponding list box． \\
\hline Print．．． & Opens the Print：General（page 99）dialog to let you print the contents of the statistics box． \\
\hline Print as PDF．．． & \begin{tabular}{l}
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 Qlik－ View＇s homepage and installed separately．
\end{tabular} \\
\hline Send to Excel & Exports possible（including selected）values to Microsoft Excel，which is auto－ matically launched if not already running．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． \\
\hline Export．．． & Opens the Save as dialog where path，file name and（table）file type for the exported data content can be specified． \\
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object． \\
Data \\
Copies the data rows of the statistics box to the clipboard． \\
Cell Value \\
Copies the text value of the statistics box cell right－clicked upon（when invok－ ing 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 lay－ out or in another document opened within the current instance of QlikView．
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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．
\end{tabular} \\
\hline Minimize & Iconizes the object．Clicking \(\square\) 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 479）page． \\
\hline Maximize & Enlarges the object to fill the screen．Clicking \({ }^{⿴ 囗}\) in the object caption（if shown）produces the same result．This command is available only if max－ imizing is allowed in the object＇s Properties dialog on the Caption（page 479）page． \\
\hline Restore & Restores a minimized or maximized object to its previous size and location． Double－clicking the icon of a minimized object or clicking 圆 in the object caption（if shown）of a maximized object produces the same result．This com－ mand is available only for minimized or maximized objects． \\
\hline Help & Opens QlikView help． \\
\hline
\end{tabular}

Remove Removes the sheet object.

\subsection*{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.


\section*{Statistics Box Properties, General}

Title
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.

\section*{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 Object ID is used for Internal Macro Interpreter (page 873) 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.

Field

Use as Title Click the button Use as Title if you want the field name as box title if not, type a title in the Title edit box. The title can also be defined as a Calculated Formula (page 851) for dynamic update of the label text. Click the ... button to open the Edit Expression dialog for easier editing of long formulas.
\begin{tabular}{ll} 
Available Functions & \begin{tabular}{l} 
Select the statistical functions you want from the list in the Available Func- \\
tions window and move them to the Displayed Functions window by double- \\
clicking or clicking the Add button.
\end{tabular} \\
Show Standard Error & \begin{tabular}{l} 
Check Show standard Error to display standard error values for Average and \\
Std dev calculations.
\end{tabular} \\
Cell Borders & \begin{tabular}{l} 
Check Cell Borders to display separation by horizontal lines, resembling the \\
rows of a table.
\end{tabular} \\
Background... & \begin{tabular}{l} 
Opens the Background Settings (page 472) dialog.
\end{tabular} \\
Displayed Functions & \begin{tabular}{l} 
Lists the statistical functions that will be used in the statistics box.
\end{tabular} \\
Settings for Selected & \begin{tabular}{l} 
The appearance of the statistics box can be further modified from the Settings \\
for Selected Row group.
\end{tabular} \\
Row & \begin{tabular}{l} 
Replace default function names by entering alternative labels here.
\end{tabular} \\
Label & \begin{tabular}{l} 
Specifies the background color of the selected row. The background color can \\
be defined as a solid color or a gradient via the Color Area dialog that opens \\
when you click the button.
\end{tabular} \\
Text Color & \begin{tabular}{l} 
This alternative lets you pick a text color for the selected row.
\end{tabular} \\
Apply Colors to all & \begin{tabular}{l} 
Check this alternative before clicking Apply or OK to apply the selected color \\
to all rows of the statistics box.
\end{tabular}
\end{tabular}

The available statistical functions are:
\begin{tabular}{ll} 
Numeric count & Sample size. Number of numeric values among the possible values) \\
Null count & Number of empty field values among the possible values \\
Text count & \begin{tabular}{l} 
Number of alpha-numeric values among the possible values
\end{tabular} \\
Total count & \begin{tabular}{l} 
Total number of possible values. This is the same number as the frequency that \\
can be shown in a list box. (sum of Numeric count and Text count)
\end{tabular} \\
Missing count & \begin{tabular}{l} 
Number of non-numeric values among the possible values (sum of Null count \\
and Text count)
\end{tabular} \\
Sum & \begin{tabular}{l} 
Sample sum
\end{tabular} \\
Average & Sample arithmetic mean (average value) \\
Std dev & Sample skewness \\
Skewness & Sample kurtosis \\
Kurtosis & Sample minimum \\
Min & Sample maximum possible numeric value \\
Max & Sample fractile
\end{tabular}

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.

\subsection*{34.3 Statistics Box Properties: Number}


\section*{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:
\begin{tabular}{ll} 
Number & \begin{tabular}{l} 
Shows numeric values with the number of digits set in the Precision spinner \\
box.
\end{tabular} \\
Integer & \begin{tabular}{l} 
Shows numeric values as integers.
\end{tabular} \\
Fixed to _Decimals & \begin{tabular}{l} 
Shows numeric values as decimal values with the number of decimal digits set \\
in the Decimals spinner box.
\end{tabular} \\
Money & \begin{tabular}{l} 
Shows values in the format set in the Format Pattern edit box. An example of \\
this format is shown in the Preview text box. The default format is the Win- \\
dows Currency setting.
\end{tabular} \\
Date & \begin{tabular}{l} 
Shows values that can be interpreted as dates in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Time & \begin{tabular}{l} 
Shows values that can be interpreted as time in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular}
\end{tabular}

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 \(=m m\) 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.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{35 Multi Box}
\begin{tabular}{|c|c|c|c|}
\hline & & & - \\
\hline OrderlD & [2) & \(\bigcirc\) & \\
\hline OrderDate & E2 & O & \\
\hline SalesPerson & [2) & O & \\
\hline Customer & E2 & O & \\
\hline Country & F2 & O & \\
\hline Shipper & 國 & O & \\
\hline
\end{tabular}

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 491) 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 489). It can also be accessed from the Object menu, when the multi box is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Multi Box Properties dialog where Multi Box Properties: General \\
(page 491), Multi Box Properties: Sort, Multi Box Properties: Presentation \\
(page 494), Multi Box Properties: Number (page 498), Font (page 475) and \\
Layout (page 476) parameters can be set.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of View \\
Menu (page 58) is activated or when the Always Show Design Menu Items \\
(page 86) 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
\end{tabular} \\
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.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Select Possible & All non-excluded values of the field are selected. \\
\hline Select Excluded & All Excluded values of the field are selected. \\
\hline Select All & All values of the field are selected. \\
\hline Clear All Selections & Clears all the current selections of the multi box. \\
\hline Clear & Clears all the current selections of the field. \\
\hline Clear Other Fields & 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. \\
\hline Lock & Locks the selected value(s) of the field. \\
\hline Lock All Selections & Locks the selected value(s) of the multi box. \\
\hline Unlock & Unlocks the locked value(s) of the field. \\
\hline Unlock All Selections & Unlocks the locked value(s) of the multi box. \\
\hline Print... & Opens the Print: General (page 99) 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). \\
\hline Print as PDF... & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Send to Excel & 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. 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). \\
\hline Export... & Opens a dialog where you can export the multi box as a table to a file of your choice. The file formats offered include a range of delimited text file formats: HTML, XML, BIFF (native Excel format) and QVD Files (page 393). \\
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Data \\
Copies the data rows of the multi box to the clipboard. \\
Cell Value \\
Copies the text value of the multi box cell right-clicked upon (when invoking the Object menu) to the clipboard. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the User Preferences dialog, Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.
\end{tabular} \\
\hline
\end{tabular}
\begin{tabular}{ll} 
Linked Objects & \begin{tabular}{l} 
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.
\end{tabular} \\
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking \\
same result. This command is available only if minimizing is allowed in the \\
object's Properties dialog on the Caption (page 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 亶 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & \begin{tabular}{l} 
Removes the sheet object.
\end{tabular}
\end{tabular}

\subsection*{35.2 Multi Box Properties: General}


\footnotetext{
Multi Box Properties, General
}
\begin{tabular}{ll} 
Title & \begin{tabular}{l} 
In the Title window the multi box can be given a name which will be dis- \\
played in the window caption. The title can also be defined as a Calculated \\
Formula (page 851) for dynamic update of the label text. Click the ... button \\
to open the Edit Expression dialog for easier editing of long formulas.
\end{tabular} \\
Alternate State & \begin{tabular}{l} 
Choose one of the available states in the list. The following Alternate States are always avail- \\
able. \\
Inherited \\
The sheets and sheet objects are always in the inherited state unless overridden by the Qlik- \\
View 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.
\end{tabular} \\
Available Fields & \begin{tabular}{l} 
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
\end{tabular} \\
(page 365) check Show System Fields. \\
Key fields will be indicated with a key symbol. Select the items to be used/re- \\
moved by clicking them. Use the Add > or the < Remove button to move
\end{tabular}

Sort by Applicability When this option is checked, the sort order of the Fields Displayed in Multibox will be dynamically updated during selections so that fields with nonexcluded 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.

Object ID
This is used for Internal Macro Interpreter (page 873) 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.

\subsection*{35.3 List Box Properties: Sort}


\section*{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.

\section*{Sort by:}
\begin{tabular}{ll} 
State & \begin{tabular}{l} 
Sorts the field values according to their logical state (selected, optional or \\
excluded).
\end{tabular} \\
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.
\end{tabular}
\begin{tabular}{ll} 
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.
\end{tabular}

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.

\subsection*{35.4 Multi Box Properties: Presentation}


\section*{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 -.
\begin{tabular}{ll} 
Alignment & \begin{tabular}{l} 
Here you set the alignment of field values. The alignment of Text and \\
Numbers is set separately.
\end{tabular} \\
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 \(n\) Lines \\
& Here the desired limit to the number of cell rows is specified.
\end{tabular}
\(\left.\begin{array}{ll}\text { Frequency } & \begin{array}{l}\text { Show } \\ \text { Toggles the status for whether the frequency of the selected field value is } \\ \text { shown or not. By frequency is meant the number of selectable combinations in } \\ \text { which the value occurs. } \\ \text { In Percent }\end{array} \\ \text { Toggles the status for whether the frequency should be shown in absolute } \\ \text { numbers or as percentages of the total number of entries. }\end{array}\right\}\)

Background... Opens the Background Settings (page 472) dialog.

\section*{Advanced Field Settings}


\section*{Advanced Field Settings}

This dialog can be accessed from Multi Box Properties: Presentation (page 494), Table Box Properties: Presentation (page 508) and Chart Properties: Dimensions (page 600). The image options described below are unavailable for bitmap charts.

\section*{Image Options}

\section*{Representation}

Hide Text When Image Missing

Selection Style Override

The following alternatives are given:

\section*{Text}

When selecting this option the expression values will always be interpreted and displayed as text.

\section*{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: \(\backslash\) 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.

\section*{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.

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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

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.

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.

\section*{Search Options}

This group allows you to control certain aspects of text search applicable to opened multi boxes and dropdown select in tables.

\section*{Include Excluded Values in Search}

\section*{Default Search Mode}

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 77) applies.
Yes
Excluded values are always included in text search.
No
Excluded vales are never included in text search, whereas optional values are included as usual.

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 \(\sim\) as part of the search string. The following alternatives are available:
<use default>
The default as specified under User Preferences (page 77) 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 ( \(\sim\) ) 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.

\subsection*{35.5 Multi Box Properties: Number}


\section*{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 434) 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:
\begin{tabular}{ll} 
Mixed & Both numbers and text. Numbers shown in original format. \\
Number & \begin{tabular}{l} 
Shows numeric values with the number of digits set in the Precision spinner \\
box.
\end{tabular} \\
Integer & Shows numeric values as integers. \\
Fixed to _Decimals & \begin{tabular}{l} 
Shows numeric values as decimal values with the number of decimal digits set \\
in the Decimals spinner box.
\end{tabular} \\
Money & \begin{tabular}{l} 
Shows values in the format set in the Format Pattern edit box. An example of \\
this format is shown in the Preview text box. The default format is the Win- \\
dows Currency setting.
\end{tabular} \\
Date & \begin{tabular}{l} 
Shows values that can be interpreted as dates in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Time & \begin{tabular}{l} 
Shows values that can be interpreted as time in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Timestamp & \begin{tabular}{l} 
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.
\end{tabular} \\
Interval & \begin{tabular}{l} 
Shows time as sequential time increment (e.g. format = mm shows the value as \\
the number of minutes since calendar start ( \(1899: 12: 30: 24: 00)\).
\end{tabular}
\end{tabular}

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 434) page where the default number format of any field can be edited.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{36 Table Box}
\begin{tabular}{|l|l|l|}
\hline Suppliers & Category & Product \\
\hline Supplier & Baby Clothes & Mehmet-Napp \\
\hline ABC & Baby Clothes & Mehmet-Skor \\
\hline ABC & Baby Clothes & Mehmet-Tröja \\
\hline ABC & Swimwear & Summer Shorts \\
\hline Asin Fashion Ltd Co & Vomen's Clothes & Okkaba Skin Jackets \\
\hline Asin Fashion Ltd Co & Vomen's Footwear & Walking Shoes \\
\hline Asin Fashion Ltd Co & Children's Clothes & RDL Suit \\
\hline Austerlich & Men's Clothes & Bow tie \\
\hline Austerlich & Men's Footwear & Davenport Shoes \\
\hline Austerlich & & \\
\hline
\end{tabular}

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 501). It can also be accessed from the Object menu, when the table box is the active object.

\subsection*{36.1 Using the Table Box}

\section*{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.

\section*{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.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Table Box Properties dialog where Table Box Properties: General \\
(page 505), Table Box Properties: Sort (page 507), Table Box Properties: Pre- \\
sentation (page 508), Chart Properties: Style (page 735), Table Box Prop- \\
erties: Number (page 511), Font (page 475), and Layout (page 476) \\
parameters can be set.
\end{tabular} \\
Fit Columns to Data & \begin{tabular}{l} 
Adjusts the width of all columns in the tables to the widest data in each col- \\
umn. The header is included in the calculation.
\end{tabular} \\
Equal Column Width & \begin{tabular}{l} 
Sets the column width in the table equal to that of the column on which you \\
clicked.
\end{tabular} \\
Sort & \begin{tabular}{l} 
Sorts the records by the field on which you click.
\end{tabular}
\end{tabular}
\begin{tabular}{|c|c|}
\hline Custom Format Cell & Opens the Custom Format Cell Dialog (page 504) which lets you format cells in the column and stripe which you clicked upon. This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated. \\
\hline Order & \begin{tabular}{l}
This cascade menu is only available when the Design Grid command of View Меnu (page 58) is activated or when the Always Show Design Menu Items (page 86) 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 .
\end{tabular} \\
\hline Change Value & 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. \\
\hline Restore Values & \begin{tabular}{l}
Only available for table box columns displaying input fields. Opens a cascade menu with three options. \\
Restore Single Value \\
Restores the value in the field value clicked on to its default value from the script. \\
Restore Possible Values \\
Restores the values of all possible field values to their default values from the script. \\
Restore All Values \\
Restores the values of all field values to their default values from the script.
\end{tabular} \\
\hline Select Possible & All non-excluded values of the field are selected. \\
\hline Select Excluded & All excluded values of the field are selected. \\
\hline Select All & All values of the field are selected. \\
\hline Clear All Selections & Clears selections in all fields displayed in the table box. \\
\hline Clear & Clears all the current selections of the field. \\
\hline Clear Other Fields & Selects all possible values in the current field, then clears the selections in all other fields. \\
\hline Lock & Locks the selected value(s) of the field. \\
\hline Unlock & Unlocks the locked value(s) of the field. \\
\hline Print... & Opens the standard Print: General (page 99) dialog, allowing you to print the table. \\
\hline
\end{tabular}
\begin{tabular}{ll} 
Print as PDF... & \begin{tabular}{l} 
Opens the Print dialog with the QlikViewPDF printer preselected. After hit- \\
ting OK 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 sys- \\
tem. The QlikViewPDF printer must be downloaded from QlikView's home- \\
page and installed separately. \\
Send to Excel \\
Exports the table to Microsoft Excel, which is automatically launched if not \\
already running. The table will appear in a new Excel worksheet. For this func- \\
tionality to work Microsoft Excel 97 or later must be installed on the com- \\
puter. \\
Export... \\
Exports the contents of the table to a file of your choice. The file formats \\
offered include a range of delimited text file formats, HTML, XML, BIFF \\
(native Excel format) and QVD (QlikView Data Files).
\end{tabular} \\
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object. \\
Full Table \\
Copies the table to the clipboard, complete with header and selection status.
\end{tabular} \\
Table Data Area \\
Copies only the values of the table to the clipboard. \\
Cell Value \\
Copies the text value of the list box cell right-clicked upon (when invoking \\
Opens QlikView help. \\
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
\end{tabular}

\section*{Remove Removes the active sheet object from the sheet.}

\section*{Custom Format Cell Dialog}


\section*{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 58) is activated or the Always Show Design Menu Items (page 86) 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 predefined styles, the custom formatting will be lost.

\section*{Note!}

Table styles in general will be superseded by any formatting resulting from attribute expressions.
\begin{tabular}{ll} 
Table Preview Pane & \begin{tabular}{l} 
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 for- \\
mats to the different groups of cells. Just click in a cell to move formatting \\
focus.
\end{tabular} \\
Undo & The Undo button can be used for undoing the changes made within the Cus- \\
tom 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.
\end{tabular}
\begin{tabular}{ll} 
Sample Text & This is the preview cell which shows the settings made. \\
Background Color & Defines the cell background color. \\
Text Color & Defines the cell text color. \\
Border before cell & \begin{tabular}{l} 
Here you define the border to precede the cell. There is a drop-down for set- \\
ting 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.
\end{tabular} \\
Border after cell & \begin{tabular}{l} 
Here you define the border to succeed the cell. There is a drop-down for set- \\
ting 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
\end{tabular} \\
look differently from the preview.
\end{tabular}\(\quad\)\begin{tabular}{l} 
In this group you fins a number of text modifiers applicable to the cell. \\
Text Size \\
Use the drop-down to apply a size modification to the general font used in the \\
table. \\
Bold \\
Mark this check box for bold text. \\
Italic \\
Mark this check box for italic text.
\end{tabular}

\subsection*{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 851) 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
\(\left.\begin{array}{ll}\text { Available Fields } & \begin{array}{l}\text { In this column the names of the data source fields are shown. Initially all fields } \\
\text { (excluding System fields) appear in this column. To include the system fields, } \\
\text { check Show System Fields. Select the items to be used/removed by clicking } \\
\text { them. Use the Add > or the < Remove button to move them to the desired col- } \\
\text { umn. }\end{array} \\
\text { Provided that the script has been executed in QlikView version } 5.03 \text { or later, } \\
\text { key fields will be indicated with a key symbol. }\end{array}\right\}\)\begin{tabular}{l} 
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 col- \\
umn. \\
Tablebox \\
When a field is selected in this list, you may give it a custom Label in the edit \\
window.
\end{tabular}\(\quad\)\begin{tabular}{l} 
Check this box to make the system fields appear in the Available Fields col- \\
umn.
\end{tabular}
\begin{tabular}{ll} 
Load Order & \begin{tabular}{l} 
Sorts the fields in the Fields Displayed in Tablebox column in load order, i.e. \\
the order in which they are read from the database.
\end{tabular} \\
Text Order & \begin{tabular}{l} 
Sorts the fields in the Fields Displayed in Tablebox column in alphabetical \\
order.
\end{tabular} \\
Alternate State & \begin{tabular}{l} 
Choose one of the available states in the list. The following Alternate States are always avail- \\
able. \\
Inherited \\
The sheets and sheet objects are always in the inherited state unless overridden by the Qlik- \\
View 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.
\end{tabular} \\
Object ID & \begin{tabular}{l} 
This is used for Internal Macro Interpreter (page 873 ) 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.
\end{tabular} \\
Calculation Condition & \begin{tabular}{l} 
Specify a Calculation Condition 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.
\end{tabular} \\
Error Messages & \begin{tabular}{l} 
The standard error messages in table boxes (and charts) can be customized in \\
the Custom Error Messages (page 855) dialog which is opened from the
\end{tabular} \\
Error Messages button.
\end{tabular}

\subsection*{36.4 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.

\section*{Sort by:}
\begin{tabular}{ll} 
Expression & \begin{tabular}{l} 
Field values will be sorted according to an arbitrary expression that is entered \\
into the text edit box below this sort option.
\end{tabular} \\
Frequency & \begin{tabular}{l} 
Toggles the status whether or not the sorting procedure of values should be \\
made by frequency.
\end{tabular} \\
Numeric value & \begin{tabular}{l} 
Toggles the status whether or not the sorting procedure of values should be \\
made by numeric value.
\end{tabular} \\
Text & \begin{tabular}{l} 
Toggles the status whether or not the sorting procedure of values should be \\
made in alphabetical order according to the ASCII standard.
\end{tabular} \\
Load order & \begin{tabular}{l} 
Toggles the status whether or not the sorting procedure of values should be \\
made in load order.
\end{tabular}
\end{tabular}

If more than one sort order is specified, the sort order will be expression, frequency, numeric, text, load order.

\subsection*{36.5 Table Box Properties: Presentation}


\section*{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.
\begin{tabular}{ll} 
Alignment & \begin{tabular}{l} 
The default alignment of field values can be set separately for Text and \\
Numbers by selecting the field in the Fields list and the appropriate option in \\
the Alignment group. As default, text values are left aligned, numbers values \\
are right aligned.
\end{tabular} \\
Omit Rows Where & \begin{tabular}{l} 
If checked, rows containing a field value that is NULL from the selected field \\
are removed. By NULL "no value" is understood.
\end{tabular} \\
Field is NULL & \begin{tabular}{l} 
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.
\end{tabular} \\
Dropdown Select & \begin{tabular}{l} 
This button opens the Advanced Field Settings (page 496) dialog which \\
offers settings for image representation of field values and special text search \\
options.
\end{tabular} \\
Advanced... & \begin{tabular}{l} 
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 TextHeader Height \(\boldsymbol{n}\) Lines
\end{tabular} \\
Mith this option selected, the header will display its contents in more than one
\end{tabular}

\subsection*{36.6 Chart Properties: Style}


Cell Background Color Transparency

\(\square\) Border Above Spacing
Background..

\section*{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.
\(\left.\begin{array}{ll}\text { Current Style } & \begin{array}{l}\text { Choose an appropriate table style from the drop-down list. If the value [Cus- } \\ \text { tom] appears in the drop-down control a custom style has been applied to the } \\ \text { table. If you change the setting back to one of the pre-defined styles, the cus- } \\ \text { tom formatting will be lost. }\end{array} \\ \text { Stripes every_Rows } & \begin{array}{l}\text { Here you can specify if and at how long intervals shaded stripes should } \\ \text { appear. }\end{array} \\ \text { Indent Mode } & \begin{array}{l}\text { This setting is only valid for pivot tables. With this alternative checked, you } \\ \text { can achieve a slightly different table style that is especially useful when you } \\ \text { need to accommodate a number of dimension labels within a limited table } \\ \text { width. } \\ \text { Use Only First Dimension Label } \\ \text { This setting is only available for pivot tables already in Indent Mode and mod- } \\ \text { ifies the style of the pivot table further. }\end{array} \\ \text { Vertical Dimension } & \begin{array}{l}\text { This setting determines whether vertical cell borders are displayed for dimen- } \\ \text { sion columns. }\end{array} \\ \text { Cell Borders } & \begin{array}{l}\text { As above, but for expression columns. }\end{array} \\ \text { Vertical Expression } & \begin{array}{l}\text { Provided that a Spacing has been determined in the Advanced Field Settings }\end{array} \\ \text { Border Above Spacing } \\ \text { Bage 496) dialog, the table style can be slightly modified by checking this }\end{array}\right\}\)
```

Cell Borders Trans- Sets how pronounced the cell borders should be. parency

```

\subsection*{36.7 Table Box Properties: Number}


\section*{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 434) 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:
\begin{tabular}{ll} 
Mixed & Both numbers and text. Numbers shown in original format. \\
Number & \begin{tabular}{l} 
Shows numeric values with the number of digits set in the Precision spinner \\
box.
\end{tabular} \\
Integer & \begin{tabular}{l} 
Shows numeric values as integers.
\end{tabular} \\
Fixed to _Decimals & \begin{tabular}{l} 
Shows numeric values as decimal values with the number of decimal digits set \\
in the Decimals spinner box.
\end{tabular} \\
Money & \begin{tabular}{l} 
Shows values in the format set in the Format Pattern edit box. An example of \\
this format is shown in the Preview text box. The default format is the Win- \\
dows Currency setting.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Date & \begin{tabular}{l} 
Shows values that can be interpreted as dates in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Time & \begin{tabular}{l} 
Shows values that can be interpreted as time in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Timestamp & \begin{tabular}{l} 
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.
\end{tabular} \\
Interval & \begin{tabular}{l} 
Shows time as sequential time increment (e.g. format \(=m m\) shows the value as \\
the number of minutes since calendar start \((1899: 12: 30: 24: 00)\).
\end{tabular}
\end{tabular}

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 434) page where the default number format of any field can be edited.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{37 Current Selections Box}
\begin{tabular}{lll}
\hline Current Selections & \\
Fields & Values \\
City & \(\circ\) & Boise \\
SupplierCo... & \(\circ\) & Asin Fashion Ltd Co, ABC
\end{tabular}

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 111) window but it is positioned directly on the sheet like any other sheet object. The Indicator (page 109) 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 513) will be displayed. It can also be accessed from the Object menu, when the current selections box is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Current Selections Box Properties dialog where Current Selec- \\
tions Box Properties: General (page 515), Font (page 475) and Layout (page \\
476) parameters can be set.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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
\end{tabular} \\
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.
\end{tabular}
\begin{tabular}{ll} 
Lock & Locks the selected value(s) of the field. \\
Unlock & Unlocks the locked value(s) of the field. \\
Print... & \begin{tabular}{l} 
Opens the Print: General (page 99) dialog where print settings can be spec- \\
ified. All values of the current selections box are sent to the printer.
\end{tabular} \\
Print as PDF... & \begin{tabular}{l} 
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 Qlik- \\
View’s homepage and installed separately.
\end{tabular} \\
Send to Excel & \begin{tabular}{l} 
Exports the text to Microsoft Excel, which is automatically launched if not \\
already running. The text will appear in the cells of a new Excel worksheet.
\end{tabular} \\
For this functionality to work Microsoft Excel 97 or later must be installed on \\
the computer.
\end{tabular}\(\quad\)\begin{tabular}{l} 
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).
\end{tabular}
\begin{tabular}{ll} 
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 圆 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & Opens QlikView help. \\
Remove & Removes the sheet object.
\end{tabular}

\subsection*{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.


\section*{Current Selections Box Properties, General}

Title

\section*{Alternate State}

The text to appear in the current selections box caption area. The title can also be defined as a Calculated Formula (page 851) for dynamic update of the label text. Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

Choose one of the available states in the list. The following Alternate States are always available.

\section*{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.

\section*{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 \begin{tabular}{l} 
This is used for Internal Macro Interpreter (page 873) 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.
\end{tabular}

In the Displayed Columns group, you determine if the status and/or the values columns will appear in the current selections box.
\begin{tabular}{ll} 
Status & \begin{tabular}{l} 
Checking this box will add a Status column with indicator to the current selec- \\
tions box.
\end{tabular} \\
Values & \begin{tabular}{l} 
Checking this box will add a Values column to the current selections box, list- \\
ing selected field value.
\end{tabular} \\
Drop-down Select & \begin{tabular}{l} 
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.
\end{tabular} \\
Clear Icons & \begin{tabular}{l} 
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.
\end{tabular} \\
Lock/Unlock Icons & \begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

Checking Use Column Labels enables the following settings:
Fields The label to be shown above the Fields column can be edited in the text box.
Status The label to be shown above the Status column can be edited in the text box
Values The label to be shown above the Values 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.
Label Background Defines the background color of the label row.
Label Text Color Defines the text color of the label row.
Text Color
Defines the text color of the display area.
Background...
Opens the Background Settings (page 472) dialog.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{38 Input Box}

\section*{Exchange Rate \(\$->€=0.761\)}

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 519). It can also be accessed from the Object menu, when the input box is the active object.

\subsection*{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 169).

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.

\subsection*{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:

\section*{Properties...}

Opens the Input Box Properties page where its parameters can be set.
\begin{tabular}{|c|c|}
\hline Order & \begin{tabular}{l}
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) 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 .
\end{tabular} \\
\hline Print... & Opens the Print: General (page 99) dialog where print settings can be specified. All values of the current selections box are sent to the printer. \\
\hline Print as PDF... & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Send to Excel & Exports the input box values to Microsoft Excel, which is automatically launched if not already running. 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. \\
\hline 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). \\
\hline Copy to Clipboard & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline
\end{tabular}
\begin{tabular}{ll} 
Minimize & \begin{tabular}{l} 
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 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 圆 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & Removes the sheet object.
\end{tabular}

\subsection*{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
Title
The text to appear in the input box caption area. The title can be defined as a Calculated Formula (page 851) for dynamic update of the label text. Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

\section*{Available Variables}

Show System Variables

\section*{New Variable}

Displayed Variables

Promote

\section*{Demote}

Label

\section*{Alternate State}

Object ID

All available variables are listed in this column. To include the system variables, mark the Show System Variables check box. Select the items to be used/removed by clicking them. Use the Add > or the \(<\) Remove button to move them to the desired column.

Makes the system variables show in the list of Available Variables.
Opens the New Variable dialog where a custom variable may be defined.
Variables to be displayed in the input box are listed in this column. Initially the column is empty.

Moves a variable upwards in the display order.
Moves a variable downwards in the display order.
An alternative name to be used as variable title in the input box, can be entered here. The label may be defined as a Calculated Formula (page 851) for dynamic update. Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

Choose one of the available states in the list. The following Alternate States are always available.

\section*{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.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

This is used for Internal Macro Interpreter (page 873) 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.

\subsection*{38.4 Input Box Properties: Presentation}


Input Box Properties, Presentation
\begin{tabular}{ll} 
Variables & \begin{tabular}{l} 
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.
\end{tabular} \\
Show Equal Sign & \begin{tabular}{l} 
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.
\end{tabular} \\
Background... & Opens the Background Settings (page 472) dialog. \\
Alignment & \begin{tabular}{l} 
The alignment of the variables can be set. The variables can be individually \\
set to left, center or right aligned.
\end{tabular} \\
Row Colors & \begin{tabular}{l} 
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 click- \\
ing 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.
\end{tabular}
\end{tabular}

\subsection*{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 171) that have been added to Displayed Variables in the Input Box Properties: General (page 521) 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 737).
Variables are normally not included in bookmarks, but here it is possible to check the option Include in

\section*{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

\section*{Standard}

Input values are not checked against any specified constraints.
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.
\begin{tabular}{ll} 
Custom & \begin{tabular}{l} 
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.
\end{tabular} \\
The previous value of the variable can be referenced by the variable name. \\
Example: \\
\(\$>=\) abct1 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.
\end{tabular}

In the Value List group it is specified if and how to present a list of values for an input box.
No List 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 A scroll control will be available to the user for the selected variable in the Scroll

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.

\section*{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.
\begin{tabular}{ll} 
Listed Values & \begin{tabular}{l} 
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.
\end{tabular} \\
Comment & \begin{tabular}{l} 
This is a commentary field where the creator of a variable can describe its pur- \\
pose and function.
\end{tabular}
\end{tabular}

\subsection*{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 434) 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:
\begin{tabular}{ll} 
Mixed & Both numbers and text. Numbers shown in original format. \\
Number & \begin{tabular}{l} 
Shows numeric values with the number of digits set in the Precision spinner \\
box.
\end{tabular} \\
Integer & Shows numeric values as integers. \\
Fixed to _ Decimals & \begin{tabular}{l} 
Shows numeric values as decimal values with the number of decimal digits set \\
in the Decimals spinner box.
\end{tabular} \\
Money & \begin{tabular}{l} 
Shows values in the format set in the Format Pattern edit box. An example of \\
this format is shown in the Preview text box. The default format is the Win- \\
dows Currency setting.
\end{tabular} \\
Date & \begin{tabular}{l} 
Shows values that can be interpreted as dates in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Time & \begin{tabular}{l} 
Shows values that can be interpreted as time in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Timestamp & \begin{tabular}{l} 
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.
\end{tabular} \\
Interval & \begin{tabular}{l} 
Shows time as sequential time increment (e.g. format = mm shows the value as \\
the number of minutes since calendar start ( \(1899: 12: 30: 24: 00)\).
\end{tabular}
\end{tabular}

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.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 529) will be displayed. It can also be accessed from the Object menu, when the button is the active object.

\subsection*{39.1 Button: Object Menu}

The button Object menu is opened by right-clicking the object. The menu commands are:
\begin{tabular}{|c|c|}
\hline Properties... & Opens the Button Properties: General (page 530) page where you can set the parameters defining the button. \\
\hline Order & \begin{tabular}{l}
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) 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 .
\end{tabular} \\
\hline Send to Excel & Exports the text to Microsoft Excel, which is automatically launched if not already running. The text will appear in a single cell in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer. \\
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Text \\
Copies the text displayed on the button to the clipboard. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the User Preferences dialog, Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.
\end{tabular} \\
\hline
\end{tabular}
\begin{tabular}{ll} 
Linked Objects & \begin{tabular}{l} 
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.
\end{tabular} \\
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking \\
same result. This command is available only if minimizing is allowed in the \\
object's Properties dialog on the Caption (page 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 亶 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & \begin{tabular}{l} 
Removes the sheet object.
\end{tabular}
\end{tabular}

\subsection*{39.2 Button Properties: General}


\section*{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 851). The text may consist of several lines, divided when you press ENTER.

\section*{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:
\begin{tabular}{ll} 
Aqua & \begin{tabular}{l} 
Default for new buttons. Gives a rounded button with three-dimensional \\
glassy look.
\end{tabular} \\
Plain & \begin{tabular}{l} 
Produces a traditional plain QlikView button. \\
System Default \\
Provides a solid background in the color defined for buttons in the operating \\
system.
\end{tabular} \\
Image & \begin{tabular}{l} 
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
\end{tabular} \\
\begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

\section*{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.

\section*{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.}

Object ID

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \(\$\). The QlikView document is always in the default state.

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.

Enable Condition The expression entered here determines the status of the button. If the expression returns 0 , the button will be disabled; if the expression returns 1 , the button will be enabled. If no expression is entered, 1 is assumed. Buttons that are disabled due to underlying status cannot be enabled by means of a condition.

Print Settings... By pressing this button you will gain access to the Print Settings dialog where margins and header/footer format can be defined. The Print Settings dialog holds two pages, Print: Layout (page 100) and Print: Header/Footer (page 101), which are identical to the last two pages of the Print dialog.

\subsection*{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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Add Action Dialog (page 532) where the actions for the object are \\
chosen. Choose the type of action in the lists. Depending on the action, dif- \\
ferent parameters for the action will appear on the Actions page. The actions \\
will be executed in the order in which they appear in the list.
\end{tabular} \\
Delete & Removes the action from the object. \\
Promote & \begin{tabular}{l} 
Moves the selected action up in the list of actions.
\end{tabular} \\
Alternate State & \begin{tabular}{l} 
Sets the state that the action refers to. This setting is only relevant for actions \\
related to selections or bookmarks.
\end{tabular}
\end{tabular}

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 851).

\section*{Selection}
\begin{tabular}{|c|c|}
\hline Select in Field & Selects the values and fields that are specified. In the Search String you can specify a search mask, e.g. \((A \mid B)\) will select both \(A\) and \(B\). \\
\hline Select Excluded & Selects the excluded values in the specified field. \\
\hline Select Possible & Selects the possible values in the specified field. \\
\hline Toggle Select & Toggles between the current selection and the specified Field and Search String. In the Search String you can specify a search mask, e.g. (A|B) will select both A and B. \\
\hline Forward & Goes one step forward in the list of selections. \\
\hline Back & Goes back one step in the list of selections. \\
\hline Pareto Select & Makes a pareto selection in the specified field based on an expression and percentage. This type of selection is used to select the top contributors to a measure, typically in line of a general 80/20 rule. For example, to find the top customers that contribute to \(80 \%\) of the turnover, Customer should be used as field, sum(Turnover) should be used as expression and 80 should be used as percentage. \\
\hline Lock Field & Locks the selections in the specified field. \\
\hline Lock All & Locks all values in all fields. \\
\hline Unlock Field & Unlocks the selections in the specified field. \\
\hline Unlock All & Unlocks all values in all fields. \\
\hline Unlock and Clear All & Unlocks all values and clears all selections in all fields. \\
\hline Clear Other Fields & Clears all fields except the one specified. \\
\hline Clear All & Clears all selections except locked ones. \\
\hline Clear Field & Clears the selection in the specified field. \\
\hline Copy State Contents & Copies the selection from the Source State to the Target State. \\
\hline Swap State Contents & Swaps the selections between State 1 and State 2. \\
\hline
\end{tabular}

\section*{Layout}

Activate Object
Activate Sheet
Activate Next Sheet
Activate Previous
Sheet
Maximize Object
Minimize Object
Restore Object
Set State Name
Bookmark

Apply Bookmark

Create Bookmark

Replace Bookmark
Print

\section*{Print Object}

Print Sheet

Print Report

Activates the object specified by Object ID.
Activates the sheet specified by Sheet ID.
Opens the next sheet in the document.
Opens the previous sheet in the document.

Maximizes the object specified by Object ID.
Minimizes the object specified by Object ID.
Restores the object specified by Object ID.
Applies the specified State to the object specified by Object ID.

Applies a bookmark specified by Bookmark ID. If two bookmarks have the same ID, the document bookmark is applied. To apply the server bookmark specify ServerlbookmarkID.

Creates a bookmark from the current selection. Specify Bookmark ID and Bookmark Name.

Replaces the bookmark specified by Bookmark ID with the current selection.

Prints the object specified by Object ID. Specify the Printer name if the object should be sent to another printer than the default printer. (not available as document and sheet trigger)

Prints the sheet specified by Sheet ID. (not available as document and sheet trigger)

Prints the report specified by Report ID. Specify the Printer name if the report should be sent to another printer than the default printer. Mark the check box Show Print Dialog if you want the Windows print dialog to be shown. (not available as document and sheet trigger)

\section*{External}
\begin{tabular}{ll} 
Export & \begin{tabular}{l} 
Exports a table containing a specific set of fields, but only those records that \\
are applicable according to the made selection are exported. Click on the \\
Setup button on the Actions page to open the Export Action Settings dialog. \\
(not available as document and sheet trigger)
\end{tabular} \\
Launch & \begin{tabular}{l} 
Launches an external program. The following settings can be configured in the \\
Actions dialog: \\
Application \\
Click on Browse.. to find the application that should be launched. (not avail- \\
able as document and sheet trigger)
\end{tabular} \\
Filename \\
Enter the path to the file that should be opened with the application specified \\
above. (not available as document and sheet trigger) \\
Parameters \\
Specify parameters for the command line from which the application is started. \\
(not available as document and sheet trigger) \\
Working Directory
\end{tabular}

\section*{Note!}

Actions that trigger other actions, so called cascading actions, may cause unforeseen consequences and are not supported!

\section*{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 883) before designing server documents with macro triggers.

\section*{Export Action Settings Dialog}


The Export Action Settings page

\section*{Selection}

In this group you set what fields should be selected for export.
\begin{tabular}{ll} 
Fields & In the Fields (page 169) box you find a list of the available fields. \\
Export Lines & \begin{tabular}{l} 
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.
\end{tabular} \\
Add \(>\) & \begin{tabular}{l} 
Adds fields to the Export Lines box.
\end{tabular} \\
< Remove & \begin{tabular}{l} 
Removes fields from the Export Lines box.
\end{tabular} \\
Promote & \begin{tabular}{l} 
Moves the selected field one step up, i.e. one step to the left in the export \\
table.
\end{tabular} \\
Demote & \begin{tabular}{l} 
Moves the selected field one step down, i.e. one step to the right in the export \\
table.
\end{tabular} \\
Multi Value (*) & \begin{tabular}{l} 
By marking a field in the Export Lines box and checking this control, you \\
allow a field to have several values in the export listing.
\end{tabular}
\end{tabular}

\section*{Export to}

In this group you can choose whether you want to export the values to a file or to the clipboard.
\begin{tabular}{|c|c|}
\hline File & If you export to a file you must mark this check box and enter the file name. If no file name is entered, the values will be exported to the clipboard. The file name may be entered as a Calculated Formula (page 851). \\
\hline Browse & Opens the Export File dialog, allowing you to browse for a file to which the values should be exported. \\
\hline Clipboard & Default setting. If no file is specified above, the values will be exported to the clipboard. \\
\hline Field Selections & If this control is checked, the export file will, for each selected field, have all the possible field values on one row, separated by tabs. \\
\hline Include Labels & If this control is checked, the first position on the line (when Field Selections is set) or the first record (when Records is set) will contain the field names. \\
\hline Records & If this control is checked, the export file will have one column for each selected field, separated by tabs. \\
\hline Append to Existing File & If this control is checked the export will be appended to the export file, if the file already exists. Labels will not be exported when appending to an existing file. If the export file does not exist, this flag has no meaning. \\
\hline
\end{tabular}

\section*{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.

\section*{Full Format}

No Thousand Separator

No Formatting

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

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

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).

\section*{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 883) before designing server documents with macro triggers.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 539) will be displayed. It can also be accessed from the Object menu, when the text object is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Text Object Properties: General (page 540) dialog where you can \\
set the parameters defining the text object.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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.
\end{tabular} \\
Print... \begin{tabular}{l} 
Opens the Print: General (page 99) dialog letting you print an image of the \\
text object.
\end{tabular} \\
Orint as PDF... \begin{tabular}{l} 
Opens the Print dialog with the QlikViewPDF printer preselected. After hit- \\
ting Print you will be prompted for a file name for the PDF output file. This \\
command is only available if a PDF printer is available on the system. The
\end{tabular} \\
QlikViewPDF printer must be downloaded from QlikTech's homepage and \\
installed separately.
\end{tabular}
\(\left.\begin{array}{ll}\text { Copy to Clipboard } & \begin{array}{l}\text { This cascade menu contains the various copy options for the sheet object. } \\
\text { Text } \\
\text { Copies the text displayed in the text object to the clipboard. } \\
\text { Image }\end{array} \\
\text { Copies an image of the sheet object to the clipboard. The image will include } \\
\text { or exclude the sheet object caption and border depending on the settings in } \\
\text { the User Preferences: Export page. } \\
\text { Object } \\
\text { Copies the entire sheet object to the clipboard for pasting elsewhere in the lay- } \\
\text { out or in another document opened within the current instance of QlikView. }\end{array}\right\}\)\begin{tabular}{l} 
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.
\end{tabular}

\subsection*{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.


\section*{Text Object Properties, General}

\section*{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.

\section*{Text}

This is where you enter the text to be displayed. The text may be defined as a Calculated Formula (page 851) for dynamic update.

\section*{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: \(\backslash \mathrm{Mypic} . j \mathrm{pg}\) ) 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.

\section*{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.

\section*{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.

\section*{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.

In this group you define how QlikView should display the text or foreground image within the text object area.

\section*{Horizontal Alignment}

The text can be horizontally left-aligned, centered or right-aligned within the text object.

\section*{Vertical Alignment}

The text can be vertically top-aligned, centered or bottom-aligned within the text object.

\section*{Image Stretch}

This setting describes how QlikView formats a foreground image to fit in the text object area. There are four alternatives.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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).

\section*{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.

\section*{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 412) dialog that opens when clicking the button.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

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 873). 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.

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 99) dialog.

\subsection*{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 532).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 545) will be displayed. It can also be accessed from the Object menu, when the line/arrow object is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & Opens the Line/Arrow Object Properties: General (page 546) dialog where \\
you can set the parameters defining the line/arrow object. \\
Order & This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out layer of the sheet objects. Valid layer numbers are -128 to 127. \\
Bring to Front \\
Sets the layout layer of the sheet object to the largest value currently used by \\
any sheet object on the current sheet. \\
Send to Back \\
Sets the layout layer of the sheet object to the smallest value currently used by \\
any sheet object on the current sheet. \\
Bring Forward \\
Increases the layout layer of the sheet object by one. Maximum value is 127. \\
Send Backward \\
Decreases the layout layer of the sheet object by one. Minimum value is -128. \\
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object.
\end{tabular} \\
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 lay- \\
out or in another document opened within the current instance of QlikView.
\end{tabular}
\begin{tabular}{ll} 
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 回 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & Removes the sheet object.
\end{tabular}

\subsection*{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.


\section*{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).

\section*{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 412) 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

Object ID

Alternate State
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.

The Object ID is used for Internal Macro Interpreter (page 873) 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.

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.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

\subsection*{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 532).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 171）．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 549）will be dis－ played．It can also be accessed from the Object menu，when the slider／calendar object is the active object．

\section*{42．1 Using a Slider／Calendar Object}

\section*{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 553）for a graphical explanation of the various parts of a slider．

\section*{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．

\section*{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：
\begin{tabular}{|c|c|}
\hline Properties... & Opens the Slider/Calendar Object Properties dialog where Slider/Calendar Object Properties: General (page 551), Slider/Calendar Object Properties: Presentation (Slider Mode) (page 553), Slider/Calendar Object Properties: Presentation (Calendar Mode) (page 554), List Box Properties: Sort (page 493), Properties: Number (page 473), Font (page 475), Layout (page 476), and Caption (page 479) parameters can be set. \\
\hline Order & \begin{tabular}{l}
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) 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 .
\end{tabular} \\
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the User Preferences: 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\square\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 slid\(\mathrm{er} / \mathrm{calendar}\) can be connected to either a field or to one or a pair of variables.


\section*{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 737) 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.
\begin{tabular}{ll} 
Min Value & \begin{tabular}{l} 
Sets the minimum value for the slider/calendar in Continuous/Numeric value \\
mode.
\end{tabular} \\
Max Value & \begin{tabular}{l} 
Sets the maximum value for the slider/calendar in Continuous/Numeric value \\
mode.
\end{tabular} \\
Static Step & \begin{tabular}{l} 
Check the box and enter a value in the edit box to specify a static step for slid- \\
er/calendar values in the Continuous/Numeric value mode.
\end{tabular} \\
Value 1 & \begin{tabular}{l} 
Shows current value for the slider in Continuous/Numeric value mode if the \\
slider/calendar is Single Value. In Multi Value mode the low value is shown.
\end{tabular} \\
Value 2 & \begin{tabular}{l} 
Shows current high value for the slider/calendar in Continuous/Numeric value \\
mode. Only relevant in Multi Value mode.
\end{tabular} \\
Override Locked Field & \begin{tabular}{l} 
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
\end{tabular} \\
selected by default for slider/calendar objects.
\end{tabular}

\subsection*{42.4 Slider/Calendar Object Properties: Presentation (Slider Mode)}


\section*{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 412) 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:
\begin{tabular}{ll} 
Major Units & The number defines the number of major units on the slider scale. \\
Labels on Every \(\boldsymbol{n}\) & Defines the density of scale label text. \\
Major Unit & \\
Minor Units per Major & Defines the number of minor units between each major unit on the slider scale. \\
Unit
\end{tabular}

Scroll arrows can be visible or hidden with the alternative Show Scroll Arrows.
Slider Style can be set as Solid or Shimmer.

\subsection*{42.5 Slider/Calendar Object Properties: Presentation (Calendar Mode)}


\section*{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.

\subsection*{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 432)).

\subsection*{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 434)).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{43 Bookmark Object}

The bookmark object is a sheet object that is used for displaying bookmarks for selections. Depending on it's configuration it can also be used for adding new bookmarks or deleting old ones. The bookmark object basically offers the options of the Bookmarks Мепи (page 61).
By right-clicking the object, the Bookmark Object: Object Menu (page 557) will be displayed. It can also be accessed from the Object menu, when the bookmark object is the active object.

\subsection*{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.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Bookmark Object Properties: General (page 558) dialog where \\
you can set the parameters defining the bookmark object.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
\\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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
\end{tabular} \\
\begin{tabular}{l} 
Sets the layout layer of the sheet object to the smallest value currently used by \\
any sheet object on the current sheet. \\
Bring Forward
\end{tabular} \\
Increases the layout layer of the sheet object by one. Maximum value is 127. \\
Send Backward
\end{tabular}
\begin{tabular}{ll} 
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include \\
or exclude the sheet object caption and border depending on the settings in \\
the User Preferences dialog, Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the lay- \\
out or in another document opened within the current instance of QlikView.
\end{tabular} \\
Linked Objects & \begin{tabular}{l} 
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.
\end{tabular} \\
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking \\
same result. This command is available only if minimizing is allowed in the \\
object's Properties dialog on the Caption (page 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page
\end{tabular} \\
479) page.
\end{tabular}

\subsection*{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

\section*{Title}

\section*{Alternate State}

Object ID

\section*{Background}

The text to appear in the bookmark object caption area. The title can be defined as a Calculated Formula (page 851) for dynamic update of the label text. Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

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.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

This is used for Internal Macro Interpreter (page 873) 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.

\section*{Color}

This can be defined as a solid color or a gradient via the Color Area (page 412) dialog that opens when you click the button.

\section*{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.
\(\left.\left.\begin{array}{ll}\text { Show Add Button } & \begin{array}{l}\text { Check this alternative to display an Add Bookmark button in the bookmark } \\
\text { object. This option is selected by default. } \\
\text { Text }\end{array} \\
& \begin{array}{l}\text { Here you may type a text to be shown on the Add Bookmark button. This } \\
\text { can be defined as a Calculated Formula (page 851) for dynamic update of the }\end{array} \\
\text { label text. Click the ... button to open the Edit Expression dialog for easier } \\
\text { editing of long formulas. If no text is typed the edit box will show <use } \\
\text { default> and the text "Add Bookmark" will be shown on the button. } \\
\text { Text Color }\end{array}\right] \begin{array}{l}\text { Sets the text color of the Add Bookmark button. }\end{array}\right\}\)\begin{tabular}{l} 
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.
\end{tabular}

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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, rightclick 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 561) will be displayed. It can also be accessed from the Object menu, when the search object is the active object.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Search Object Properties: General (page 562) dialog where you \\
can set the parameters defining the search object. \\
Order \\
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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
\end{tabular} \\
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.
\end{tabular}
\begin{tabular}{ll} 
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking -0 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 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog, on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 回 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & Removes the sheet object.
\end{tabular}

\subsection*{44.2 Search Object Properties: General}

On the General tab you can configure the search options.


Search Object Properties, General
All Fields Mark this radio button to allow the search object to search all fields in the document.

\section*{List of Fields}

\section*{Selected Fields}

Mark this radio button and list the fields in the field below to set which fields should be searchable. Use semicolon (;) as separator between fields. The wildcard characters * and ? are allowed in field names. The list may be defined as a calculated formula for dynamic update.

Mark this radio button and choose what fields should be searchable.
\begin{tabular}{|c|c|}
\hline Show Fields from Table & \begin{tabular}{l}
This drop-down menu shows the value All Tables which contains all fields of the document. By selecting a specific table name in the drop-down menu you may limit the Selected Fields list to only fields from that internal table. \\
Finally, you may select the value All Tables (Qualified) which will show all fields in the document qualified with their table name. A field will appear one time for each table it appears in.
\end{tabular} \\
\hline Default Search Mode & \begin{tabular}{l}
Specifies the initial default search mode to be used in text searches. The mode can always be changed on the fly by typing * or \(\sim\) as part of the search string. The following alternatives are available: \\
<use default> \\
The default as specified under User Preferences (page 77) 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 ( \(\sim\) ) 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 Associated Search \\
The search box will open using the Associated Search.
\end{tabular} \\
\hline Object ID & The unique ID of the current search object. Upon creation, every sheet object is assigned a unique ID for control via Automation. The ID consists of a combination of letters defining the type of object, as well as of a number. The first search object of a document will be assigned the ID SO01. 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. \\
\hline Alternate States & \begin{tabular}{l}
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.
\end{tabular} \\
\hline
\end{tabular}

This drop-down menu shows the value All Tables which contains all fields of the document. By selecting a specific table name in the drop-down menu you may limit the Selected Fields list to only fields from that internal table. Finally, you may select the value All Tables (Qualified) which will show all
 Specifies the initial default search mode to be used in text searches. The mode can always be changed on the fly by typing * or \(\sim\) as part of the search string. following alternatives are available:

The default as specified under User Preferences (page 77) will apply.

\section*{rch} to facilitate a wildcard search.

\section*{Use Fuzzy Search}

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

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

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 bjects 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 booknak

Choose one of the available states in the list. The following Alternate States are always available.

The sheets and sheet objects are always in the inherited state unless overridden by the Qlikgets the same settings as the sheet if inherited is chosen.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

\subsection*{44.3 Search Object Properties: Presentation}


Search Object Properties, Presentation

\section*{Search Result}

Include Excluded Mark this check box to include those values in the search that have been excluded by selections.

Highlight Search Sub- The search string so far will be highlighted in the matches.
string
Collapse when result in Limit the number of displayed matches in each field.
field exceeds

\section*{Visual Style}

\section*{Appearance}

Show Reflection
Text in Search Field

Choose between a Rounded and Squared look.
A reflection of the search object will be shown in the layout.
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 737) is opened for easier editing of long formulas.

\subsection*{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.
\begin{tabular}{ll} 
Text & Sorts the fields containing search hits in alphabetical order. \\
Number of Hits & Sorts the fields containing search hits according to the number of hits in each \\
field.
\end{tabular}

As Listed on General Tab

Sorts the fields containing search hits according to the order in which the fields are listed on the General tab.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{45 Container}
\begin{tabular}{ll|l|}
\hline Container Object \\
\hline Table Box & & \\
\hline \begin{tabular}{ll} 
Author & \\
\hline
\end{tabular} & Title \\
\hline Adams, Douglas & Hitchikers Guide to the Galaxy, The \\
\hline Adams, Douglas & Restaurant at the End of the Universe, ... \\
\hline Adams, Douglas & So Long and Thanks for the Fish \\
\hline Adams, Douglas & Titanic \\
\hline Benson, Amber & Death's Daughter \\
\hline Briggs, Patricia & Blood Bound \\
\hline Briggs, Patricia & Bone Crossed \\
\hline Briggs, Patricia & Iron Kissed \\
\hline Briggs, Patricia & Moon Called \\
\hline Eddings, David & Pawn of Prophecy \\
\hline Eddings, David & Redemption of Althalus, The \\
\hline Eddings, David & Seeress of Kell, The \\
\hline Eddings, David & Wizard's Endgame \\
\hline Evanovich, Janet & Four to Score \\
\hline Evanovich, Janet & One for the Money \\
\hline Evanovich, Janet & Ten Big Ones \\
\hline
\end{tabular}

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 567) will be displayed. It can also be accessed from the Object menu, when the container is the active object.

\subsection*{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+RETURN.

Order
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127 .

\section*{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.

\section*{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.

\section*{Bring Forward}

Increases the layout layer of the sheet object by one. The maximum value is 127.

\section*{Send Backward}

Decreases the layout layer of the sheet object by one. The minimum value is 128.
\begin{tabular}{ll} 
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include \\
or exclude the sheet object caption and border depending on the settings in \\
the User Preferences dialog, Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the lay- \\
out or in another document opened within the current instance of QlikView.
\end{tabular} \\
Linked Objects & \begin{tabular}{l} 
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.
\end{tabular} \\
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking \\
same result. This command is available only if minimizing is allowed in the \\
object's Properties dialog on the Caption (page 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page
\end{tabular} \\
479) page.
\end{tabular}

\subsection*{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 851) for dynamic update of the label text. Click the ... button to open the Edit Expression dialog for easier editing of long formulas.


\section*{Container Properties, General}

\section*{Existing Objects}

\section*{Filter}

Objects Displayed in Container

Promote

\section*{Demote}

\section*{Object ID}

\section*{Alternate State}

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.

With this setting it is possible to filter the list of Existing Objects by, for example, object type.

A list of the objects already in the container object.

Moves the selected object one step up.
Moves the selected object one step down.
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.

Choose one of the available states in the list. The following Alternate States are always available.

\section*{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.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

\section*{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 851) for dynamic update. By clicking the ... button the full Edit Expression dialog is opened for easier editing of long formulas.

\subsection*{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.
Appearance Here you set how the different objects in the container should be displayed.
Only available in Single Object mode.
Tabs at top
All objects in the container are displayed as tabs at the top of the container.

\section*{Tabs on the left}

All objects in the container are displayed as tabs on the left hand side of the container.
Tabs to the right
All objects in the container are displayed as tabs on the right hand side of the container.

\section*{Tabs at bottom}

All objects in the container are displayed as tabs in the lower part of the container.
Dropdown at top
All objects in the container are displayed in a drop-down menu at the top of the container.

\section*{Display Object Type Icons}

Tab Color

Columns
Rows
Spacing
Background...

Enable this setting to have icons symbolizing the different objects in the tabs. Only available in Single Object mode.

Specifies the color of the tabs. The color can be defined as a solid color or a gradient via the Color Area dialog that opens when clicking the button. Only available in Single Object mode.

Select number of columns. Only available in Grid mode.
Select number of rows. Only available in Grid mode.
Select the size of the spacing. Only available in Grid mode.
Opens the Background Settings (page 472) dialog.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 573) 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 873).
The QlikView installation CD 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 the needs. (I.e. QlikTech does not offer support on these sample controls.)

\subsection*{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:

Properties for Replacement Control

Properties...
Order

\section*{Copy to Clipboard}

Opens the Properties dialog for the OCX replacement control used. This dialog is provided with the OCX replacement control and may be absent.

Opens the Custom Object Properties: General (page 574) dialog.
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127 .

\section*{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.

\section*{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.

\section*{Bring Forward}

Increases the layout layer of the sheet object by one. Maximum value is 127 .

\section*{Send Backward}

Decreases the layout layer of the sheet object by one. Minimum value is -128 .
This menu contains the various copying options for the sheet object.
Image
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 User Preferences dialog, Export page.

\section*{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.
\begin{tabular}{ll} 
Linked Objects & \begin{tabular}{l} 
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.
\end{tabular} \\
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking \\
same result. This command is available only if minimizing is allowed in the \\
object's Properties dialog on the Caption (page 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 回 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & \begin{tabular}{l} 
Removes the sheet object.
\end{tabular}
\end{tabular}

\subsection*{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.


\footnotetext{
Custom Object Properties, General
}

Select OCX Replacement OCX Properties

\section*{Alternate State}

\section*{Currently Selected OCX \\ The name of the currently selected OCX replacement.}

Object ID This is used for Internal Macro Interpreter (page 873) 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.

Click this button to open the Insert Object dialog where you may select or change an OCX replacement control.

Opens the Properties dialog for the OCX replacement control used. This dialog is programmed by the provider of the OCX replacement control and may vary in looks and functionality. It may also be unavailable for certain replacement controls.

Choose one of the available states in the list. The following Alternate States are always available.

\section*{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.

\section*{Default state}

This is the state where most QlikView usage occurs and is represented by \$. The QlikView document is always in the default state.

\subsection*{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 476).

\subsection*{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 479).

\section*{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

\subsection*{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.

\subsection*{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.

\subsection*{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.
For information about a more list-oriented way to manage your personal server objects, see the Server Objects Dialog (page 578).

\section*{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.
\begin{tabular}{ll} 
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.
\end{tabular}

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.

\subsection*{47.4 Server Objects Dialog}

The Server Objects dialog is used to manage your personal server objects and to select among other users' shared server objects. The dialog contains two pages:

\section*{My Server Objects}


Server Objects, My Server Objects
Here you can manage your own personal server objects and share them to other users.
At the top of the dialog you will find a list of all personal server objects currently defined for the QlikView document. The objects are described by and may be sorted by several columns:
\begin{tabular}{ll} 
Show & \begin{tabular}{l} 
Mark this check box if you want the object to appear in the layout of the doc- \\
ument. If the check box is left unchecked, the object will not appear, but will \\
remain accessible via this dialog.
\end{tabular} \\
ObjectID & \begin{tabular}{l} 
The unique ID of the object. Linked objects share the same ID.
\end{tabular} \\
SheetID & The ID of the sheet where the object will reside when shown. \\
Type & The type of the object (e.g. list box). \\
Caption & The caption of the object (if defined). \\
User & \begin{tabular}{l} 
The Windows identity of the user who created the object.
\end{tabular} \\
Aodified & \begin{tabular}{l} 
This check box is only available in the My Server Objects page. By marking \\
it for one of your personal server objects, you will make that object available
\end{tabular} \\
to other authenticated users of the same server document in their Shared \\
Objects page. The object will stay in your My Server Objects page and will \\
not appear in your Shared Objects page. It is possible to revoke the sharing, at \\
any time, by deselecting the check box.
\end{tabular}

At the bottom of the dialog you will find buttons which can be used to perform actions with the objects in the list above.
\begin{tabular}{ll} 
Clear All & Removes all objects in the list. \\
Remove & Removes the selected object from the list.
\end{tabular}

\section*{Shared Objects}


Server Objects, Shared Objects
Here you can manage the display of other users' shared server objects.
At the top of the dialog you will find a list of all shared server objects currently defined for the QlikView document. The objects are described by and may be sorted by several columns:
\begin{tabular}{ll} 
Show & \begin{tabular}{l} 
Mark this check box if you want the object to appear in the layout of the \\
document. If the check box is left unchecked, the object will not appear, \\
but will remain accessible via this dialog.
\end{tabular} \\
ObjectID & The unique ID of the object. Linked objects share the same ID. \\
SheetID & The ID of the sheet where the object will reside when shown. \\
Type & The type of the object (e.g. list box). \\
Caption & The caption of the object (if defined). \\
User & The Windows identity of the user who created the object (yourself). \\
Modified & A timestamp indicating when the object was last modified.
\end{tabular}

\section*{48 Layout Themes}

\subsection*{48.1 About QlikView Layout Themes}

\section*{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.

\section*{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 583), 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 583) to extract selected properties from that object and insert them into a theme file

\section*{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.

\section*{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.

\subsection*{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

\section*{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.

\section*{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.

\section*{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.

\subsection*{48.3 Theme Maker Wizard}

Step 1 - Select Theme File
Step 2 - Source Selection
Step 3 - Select Specific Properties
Step 4 - Insertion of Properties in Theme
Step 5 - Save Theme
page 583
page 583
page 584
page 584
page 584

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.

\section*{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.
\begin{tabular}{ll} 
New Theme & \begin{tabular}{l} 
Choose this option to create a new theme. \\
Template
\end{tabular} \\
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 Browse... option for browsing for \\
theme files in other locations is found.
\end{tabular}

Click Next to continue. The Save As dialog will appear when creating a new theme.

\section*{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.
\begin{tabular}{ll} 
Source & \begin{tabular}{l} 
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
\end{tabular} \\
will be pre-selected.
\end{tabular}

\section*{Property Groups}

Click Next to continue.

\section*{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.

\section*{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.

\section*{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 Mark this check box to use this theme as default theme in the current docthis document ument. 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 Mark this check box to use this theme as default theme for new documents. new document 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.

\section*{Part 6 Charts}

\section*{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.

\section*{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


Combo chart


Line chart


Pie chart


Scatter chart


Radar chart


Grid chart


Block chart


Pivot table


Gauge chart


Funnel chart


Straight table


\section*{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.

\section*{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 592) for details.

\section*{Table Charts}


The pivot table above shows some of the more common components of a QlikView table chart.

\subsection*{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.

\subsection*{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 593)).

\section*{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.

\section*{Note!}

Selections made directly in line charts and bar charts (not combo chats) 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.

\section*{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.

\section*{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 1 st dimension. Clicking a second time (i.e. within the selection) selects a single value in the 2 nd 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 1 st dimension, all sub-values of the 2 nd and 3 rd dimensions belonging to these values are selected, not just the ones touched by the selection.

\section*{Selections in Gauge Charts}

Selections cannot be made in gauge charts.

\section*{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.

\section*{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.

\section*{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.

\subsection*{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.


\subsection*{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.

\subsection*{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 451)).
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.

\subsection*{49.6 Chart Properties: General}


\section*{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.
\begin{tabular}{ll} 
Window Title & \begin{tabular}{l} 
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 737).
\end{tabular} \\
Show Title in Chart \(\quad\)\begin{tabular}{l} 
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 for- \\
mula. Click on the ... button to open the Edit Expression Dialog (page 737). \\
The chart title is not displayed in pivot tables or straight tables.
\end{tabular} \\
Title Settings & \begin{tabular}{l} 
Define advanced settings for the chart title by clicking the Title Settings but- \\
ton.
\end{tabular}
\end{tabular}
\begin{tabular}{|c|c|}
\hline Print Settings & Clicking the Print Settings button takes you to the Print Settings dialog where it is possible to define margins and header/footer format. The Print Settings dialog holds two pages, Print: Layout (page 100) and Print: Header/Footer (page 101). \\
\hline Alternate State & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Object ID & The Object ID is used for Internal Macro Interpreter (page 873) purposes. Every sheet object is assigned a unique ID, which for charts start with CH 01. Linked objects share the same object ID. It is possible to edit this ID number later on. \\
\hline Detached & If enabled, the chart will be detached, that is, it will no longer be dynamically updated when selections are made. \\
\hline Read Only & If enabled, the chart will be read only, that is, selections cannot be made by clicking or painting with the mouse in the chart. \\
\hline Calculation Condition & 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 Edit Expression Dialog (page 737). \\
\hline Chart Type & The Chart Type group is where you select the basic layout of the chart. For more information on each chart type, see Chart Types (page 595). \\
\hline Fast Type Change & \begin{tabular}{l}
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. \\
Allowed Types \\
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. \\
Preferred Icon Position \\
In graphical charts, the fast type change icon can be positioned either inside the chart or in the sheet object caption. In table charts, the caption is the only alternative.
\end{tabular} \\
\hline Reset User Sizing & By pressing this button, all user sizing of legend, title etc. in graphical charts will be reset. Docking of individual items will not be affected. \\
\hline Reset User Docking & By pressing this button, all user docking of legend, title etc. in graphical charts will be reset. \\
\hline Error Messages & Opens the Custom Error Messages (page 855) dialog. \\
\hline 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. \\
\hline
\end{tabular}

\section*{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 593) page.
A brief introduction to chart types:
\begin{tabular}{|c|c|}
\hline Bar Chart & 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. \\
\hline Line Chart & 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 Chart Properties: Expressions (page 609)). \\
\hline Combo Chart & 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. \\
\hline Radar Chart & 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. \\
\hline Scatter Chart & The scatter chart plots data points representing combinations of expressions, iterated over one or several dimensions. Both axes are continuous, representing one expression each. \\
\hline Grid Chart & 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. \\
\hline Pie Chart & Shows the relation between a single Fields (page 169) (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 Chart properties: Expressions page, the first in the expression list will be displayed. To switch expression use the Promote/Demote buttons in the Expressions property page. \\
\hline Block Chart & 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 Color Functions (page 361) is used for creating a so called "heat chart". \\
\hline Funnel Chart & The funnel chart is typically used for showing data in flows and processes. From a display standpoint it is related to the pie chart. The chart may be shown with either segment height/width or segment area proportional to data. It is also possible to draw the chart with equal segment heights/widths without regards to data points. \\
\hline Gauge Chart & Gauge charts are used to display the value of a single expression, lacking dimensions. \\
\hline
\end{tabular}

Pivot Table

Straight Table

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.

The straight table differs from the pivot table in that it can not display subtotals and that the grouping of dimensions is shown in record form so that each row of the table contains field and expression values.

\section*{Title Settings}


The Title Settings dialog
The layout of the chart title is controlled by the various settings of this dialog.
Title Style Sets the basic style for the title.
Background Color Sets the color of the title background. The color can be defined as a solid color or a gradient via the Color Area dialog that opens when clicking the button.

Horizontal Alignment Specifies how the title text is positioned in relation to the plot area.
Font
Sets the font in the chart title. The standard Font dialog is opened when you click the button.

\section*{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 593) page.
The quickest way to create a new bar chart is to select Quick Chart Wizard (page 747) from the Tools menu.
By right-clicking the object, the Bar Chart: Object Menu (page 597) 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 595).

\subsection*{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.

\subsection*{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:

\section*{Properties... \\ Opens the Properties dialog where the parameters defining the chart can be} set.
\begin{tabular}{ll} 
Detach & \begin{tabular}{l} 
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 com- \\
parisons between the copy and the original. \\
Attach \\
Attaches a detached chart. The chart becomes dynamically linked to the data. \\
The command is available only if the chart is detached. \\
Set Reference \\
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the
\end{tabular} \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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 doc-
\end{tabular}
\begin{tabular}{ll} 
Print as PDF... & \begin{tabular}{l} 
Opens the Print dialog with the QlikViewPDF printer as default. \\
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 Qlik- \\
View's homepage and installed separately.
\end{tabular} \\
Send Values to Excel \(\quad\)\begin{tabular}{l} 
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.
\end{tabular} \\
Export... \begin{tabular}{l} 
Opens a dialog for saving an image of the chart to file. The image can be \\
saved as png, jpg, bmp or gif.
\end{tabular} \\
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table.
\end{tabular} \\
Image \\
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 User Preferences: Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the lay- \\
out or in another document opened within the current instance of QlikView.
\end{tabular}

\subsection*{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 593).

\subsection*{50.4 Chart Properties: Dimensions}


\section*{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 803).
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.

\section*{Available Fields/Groups}

\section*{Show Fields from} Table

Edit Groups...

Animate...

Trellis...

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 Hierarchic Groups (Drill-Down) (page 853) or a curved arrow for Non-Hierarchic Groups (Cyclic) (page 854). Groups are defined in the Document Properties: Groups (page 427) page. 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 number of dimensions that can be displayed varies with different chart types, as outlined in the Chart Properties: General (page 593) 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.

Checking this option will display the System Fields (page 365) in the Available Fields/Groups column.

From here, you control what fields/groups appear in the Available
Fields/Groups 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 256) fields in the load script.) It is also possible to view the fields of one table at a time. Note that available groups are always listed.

This button takes you directly to the Document Properties: Groups (page 427) page, where field groups to be used as dimensions can be defined.

Opens The Animation Dialog (page 604), 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.

Opens the Trellis Settings (page 605) 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.

\section*{Used Dimensions}

\section*{Add calculated dimen-} sion...

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.

\section*{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 361)). If the result of the expression is not a valid color representation, the program will default to black.

\section*{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 361)). If the result of the expression is not a valid color representation, the program will default to black.

\section*{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 ' \(<\mathrm{B}>\) ' for bold text, \(\quad<\mathrm{I}>\) ' for italic text and/or ' \(<\mathrm{U}>\) ' for underlined text.

With the Promote and Demote buttons, dimensions in the Used dimensions list can be sorted.

Adds a new dimension and opens it for editing in the Edit Expression Dialog (page 737) 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 289) function can be included for achieving nested aggregation.

Edit.. Opens the dimension for editing in the Edit Expression Dialog (page 737) dialog. See Add calculated dimension... above for details on calculated dimensions.

\section*{Settings for Selected} Dimension

In this group you find settings for individual dimensions.

\section*{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.

\section*{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.

\section*{Show Legend}

When Show Legend is checked, the "names" of field values are shown along the x -axis.

\section*{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 851) for dynamic update of the label text. Click on the ... button to open the Edit Expression Dialog (page 737) for easier editing of long formulas.

\section*{Advanced...}

This button opens the Advanced Field Settings (page 496) dialog which offers settings for image representation of field values and special text search options.

\section*{Comment}

A commentary field where the selected dimension can be described. The comment may be entered as a calculated formula. Click on the ... button toopen the Edit Expression Dialog (page 737).

\section*{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:

\section*{No Breaks}

Will only insert page breaks at the end of each page, as required.

\section*{Conditional Breaks}

Inserts a page break, unless all rows with the following dimension value can be fitted on the current page.

\section*{Forced Breaks}

Inserts a page break whenever the dimension value changes

\section*{The Animation Dialog}


\section*{Animate First Dimen-} sion

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.

\section*{Time Between Values} (ms)

\section*{Frames per Second}

Autoplay

Loop

Sets the time in milliseconds between each value in the animation dimension. This value may be given as a Calculated Formula (page 851).

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 851).

Enable this check box if an animation is to start automatically whenever a selection is made in the document.

Enable this option if you want the animation to play repeatedly until stopped with the Pause button in the Animation Bar.
\begin{tabular}{ll} 
Play Once & \begin{tabular}{l} 
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.
\end{tabular} \\
Show Animation & By enabling this check box, the value of the data will be shown in the chart dur- \\
ing the animation. \\
Dimension Value \\
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.
\end{tabular}

\section*{Trellis Settings}

\begin{tabular}{ll} 
Enable Trellis Chart & \begin{tabular}{l} 
Enable this check box to create an array of charts based on the chart's first \\
dimension.
\end{tabular} \\
Enable Secondary Trel- \\
lis Dimension & \begin{tabular}{l} 
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 dis- \\
played as rows in the trellis matrix.
\end{tabular} \\
Number of Columns & \begin{tabular}{l} 
Choose Auto to let QlikView decide how many columns to display or choose \\
Fixed to set the number yourself.
\end{tabular} \\
Number of Rows & \begin{tabular}{l} 
Choose Auto to let QlikView set the number of rows to display or choose \\
Fixed to set the number yourself
\end{tabular}
\end{tabular}

\subsection*{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.


\section*{Chart Properties, Dimension Limits}

This property page is used for defining dimension limits. Each dimension in the chart is configured separately.

\section*{Limits}

\section*{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.

\section*{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 737).

\section*{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 Percentage 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 737).

\section*{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 Percentage 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 737). Select Include Boundary Values to include the dimension value that contains the comparison value.

\section*{Options}

\section*{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.

\section*{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.

\section*{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.

\section*{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.
\begin{tabular}{|c|c|c|c|c|c|}
\hline \multicolumn{3}{|l|}{Expression Total} & \multicolumn{3}{|l|}{Dimension Total} \\
\hline Region & \(\triangle\) Product Family & Sales & Region & \(\triangle\) Product Family & Sales \\
\hline & & 12,250,605 & Overall Total & & 12,250,605 \\
\hline Europe & Women's Clothes & 3,138,666 & Europe & Total & 7,753,390 \\
\hline Europe & Men's Footwear & 1,280,513 & Europe & Women's Clothes & 3,138,666 \\
\hline Europe & Sportswear & 1,187,870 & Europe & Men's Footwear & 1,280,513 \\
\hline Europe & Men's Clothes & 582,220 & Europe & Sportswear & 1,187,870 \\
\hline Europe & Women's Footwear & 566,168 & Europe & Men's Clothes & 582,220 \\
\hline Europe & Baby Clothes & 563,183 & Europe & Women's Footwear & 566,168 \\
\hline Europe & Children's Clothes & 315,448 & Europe & Baby Clothes & 563,183 \\
\hline Europe & Swimwear & 119,322 & Europe & Children's Clothes & 315,448 \\
\hline North America & Women's Clothes & 707,181 & Europe & Swimwear & 119,322 \\
\hline North America & Sportswear & 423,914 & North America & Total & 2,329,970 \\
\hline North America & Women's Footwear & 333,858 & North America & Women's Clothes & 707,181 \\
\hline North America & Men's Footwear & 307,859 & North America & Sportswear & 423,914 \\
\hline North America & Men's Clothes & 217,669 & North America & Women's Footwear & 333,858 \\
\hline North America & Children's Clothes & 153,353 & North America & Men's Footwear & 307,859 \\
\hline North America & Baby Clothes & 142,239 & North America & Men's Clothes & 217,669 \\
\hline North America & Swimwear & 43,896 & North America & Children's Clothes & 153,353 \\
\hline Scandinavia & Women's Clothes & 286,116 & North America & Baby Clothes & 142,239 \\
\hline Scandinavia & Sportswear & 164,486 & North America & Swimwear & 43,896 \\
\hline
\end{tabular}

Expression Totals and Dimension Totals
When Dimension Totals is used, it is possible to have sub-totals within a straight table.

\subsection*{50.6 Chart Properties: Expressions}


\section*{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).

\section*{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.

\section*{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 \(\left.a^{\prime}+'\right)\) 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:

\section*{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 361). 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.

\section*{Text Color}

An auxiliary expression for Text Color can be created using the same method as for the background color (see above).

\section*{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 735).) The expression used as text format expression should return a string containing a ' \(<\mathrm{B}>\) ' for bold text, \(\quad<\mathrm{I}>\) ' for italic text and/or \(\quad<\mathrm{U}>\) ' for underlined text. Note that \(=\) is necessary before the string.

\section*{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.

\section*{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.

\section*{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 \(<\mathrm{W} n>\) 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: \(<\mathrm{W} 2.5\rangle\). The style of the line can be controlled by including a tag \(\langle\mathrm{S} n\rangle\) 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 \(<\mathrm{W} n>\) and \(<\mathrm{Sn}\rangle\) tags can be freely combined, but only the first occurrence of each counts. The tags must be enclosed by single quotations.

\section*{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

Delete

Copy

Export...

\section*{Paste}

New expressions and sub-expressions, are created by means of the Add button which takes you to the Edit Expression Dialog (page 737). The option is also available in the context menu that appears when right-clicking in the list of expressions.

The Delete button lets you remove previously created expressions from the list. The option is also available in the context menu that appears when rightclicking on an expression in the list of expressions.

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.

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.

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.
\begin{tabular}{|c|c|}
\hline Import & The Import 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. \\
\hline \begin{tabular}{l}
Promote \\
Demote
\end{tabular} & If several expressions are displayed, they can be sorted by means of the Promote and Demote buttons. This affects the order in which columns etc. are displayed in the chart. \\
\hline \multirow[t]{2}{*}{Group} & The Group button can be used for merging expressions into one or more cyclic groups, provided that two or more expressions are available. In the QlikView layout, you can cycle through the expressions belonging to one group by clicking the cycle icon that is displayed in the chart (= Cycle Group). 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. \\
\hline & \begin{tabular}{l}
Note! \\
Do not confuse Cycle Group with Non-Hierarchic Groups (Cyclic) (page 854)!
\end{tabular} \\
\hline Ungroup & Selecting an expression belonging to a group and clicking Ungroup, 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. \\
\hline Enable & Disabling this check box will set the expression to be omitted from the chart. \\
\hline Relative & 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. \\
\hline Invisible & Enabling this check box prevents the plotting of this expression while retaining the space allocated for it. \\
\hline Label & In front of the expression label one or several icons are used for indicating the Chart Types (page 595) used and/or the Display Options selected for the expression (see below). \\
\hline Definition & Shows the composition of the selected expression. It is possible to edit the expression directly in this box. By clicking the ... button the full Edit Expression dialog is opened. \\
\hline Comment & This is a commentary field where the creator of the expression can describe the purpose and function of the expression. \\
\hline
\end{tabular}

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.

\section*{Bar}

Shows the values of the selected expression as bars. This option is only available for bar and combo charts.

\section*{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.

\section*{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.

\section*{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 forth 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.

\section*{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 expression define a median, an upper whisker, and a lower whisker. A common extension to a Box Plot is so called outliners 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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.

\section*{Representation}

This option is only available for straight tables and pivot tables.

\section*{Text}

The expression values are always interpreted and displayed as text.

\section*{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: \(\backslash\) 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.

\section*{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 702) dialog that is reached from the Gauge Settings button.

\section*{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 468) dialog. This option is only available for straight tables.

\section*{Note!}

The mini chart will not be displayed when exporting to Excel!

\section*{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.

\section*{Examples:}
=Name \& '<url>' \& Link
=Name \& '<url>www.qlikview.com'
where Name and Link are table fields loaded in the script.

\section*{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:

\section*{No Stretch}

If this option is selected, the image will be shown as is, without any stretch-

\section*{Accumulation}

Total Mode

Bar Border Width

\section*{Expressions as Legend}
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.

\section*{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.

\section*{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.

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 yvalue 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.

\section*{No Accumulation}

If this option is selected, the \(y\)-values of the selected chart expression will not be accumulated.

\section*{Full Accumulation}

If this option is selected, each \(y\)-value will accumulate all previous \(y\)-values of the expression. See above under Accumulation.

\section*{Accumulate \(\boldsymbol{n}\) 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.

This group is enabled for the selected expression. There are three possible settings:

\section*{No Totals}

Totals are not calculated for the selected expression.

\section*{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.

\section*{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.

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).

When several expressions are used, this option displays a legend showing the expressions and their corresponding colors next to the chart.

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 \(\mathrm{y}=\mathrm{f}(\mathrm{x})\). 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.

\section*{Average}

The average is plotted as a straight line.

\section*{Linear}

A linear regression line is plotted.

\section*{Polynomial of 2nd degree}

A polynomial trend line of the second degree is plotted.

\section*{Polynomial of 3rd degree}

A polynomial trend line of the third degree is plotted.

\section*{Polynomial of 4th degree}

A polynomial trend line of the fourth degree is plotted.

\section*{Exponential}

An exponential trend line is plotted.

\section*{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.

\section*{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.

\subsection*{50.7 Chart Properties: Sort}


\section*{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.
\(\left.\begin{array}{ll}\text { Y-value } & \begin{array}{l}\text { Dimension values will be sorted by the numeric value of the y-axis. This } \\
\text { option is not available for calculated dimensions. }\end{array} \\
\text { State } & \begin{array}{l}\text { Dimension values will be sorted according to their logical state, i.e. selected } \\
\text { values before optional values, before excluded values. }\end{array} \\
\text { Expression } & \begin{array}{l}\text { Dimension values will be sorted according to the expression that is entered } \\
\text { into the text edit box below this sort option. }\end{array} \\
\text { Frequency } & \begin{array}{l}\text { Dimension values will be sorted according to the number of occurrences in the } \\
\text { table. }\end{array} \\
\text { Numeric Value } & \text { Dimension values will be sorted according to their numeric value. }\end{array}\right\}\)\begin{tabular}{l} 
Text \\
Load Order
\end{tabular}

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 \(\mathbf{A}->\mathbf{Z}\) and \(\mathbf{Z}->\mathbf{A}\).

By clicking the Default button, dimension values will be set to the default defined in the Document Properties: Sort (page 432) 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.

\subsection*{50.8 Chart Properties: Style}


\section*{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.

\footnotetext{
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

3D View

In this group the mode is set, either Grouped or Stacked mode for bars (Overlayed 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.
\begin{tabular}{lll} 
Dimensions & Expressions & Subtype \\
\(\mathbf{1}\) & 1 & Single bar \\
\(\mathbf{1}\) & 2 or more & Expressions are grouped or stacked \\
\(\mathbf{2}\) & 1 & Dimensions are grouped or stacked \\
\(\mathbf{2}\) & 2 or more & \begin{tabular}{l} 
Dimensions are grouped \\
\(\mathbf{3}\)
\end{tabular} \\
\(\mathbf{3}\) & 2 or more & \begin{tabular}{l} 
1st and 2nd dimensions are grouped, 3rd dimen- \\
sion is stacked
\end{tabular} \\
\begin{tabular}{l}
1 st and 2nd dimensions are grouped, expressions \\
are stacked
\end{tabular} \\
\(\mathbf{4}\) & 2 or more & \begin{tabular}{l}
1 st and 2nd dimensions are grouped, 3rd dimen- \\
sion is stacked \\
1 st and 2nd dimensions are grouped, expressions \\
are stacked.
\end{tabular} \\
\(\mathbf{4}\) & &
\end{tabular}

The settings in this group define the angle from which the chart is viewed in 3D modes.

\section*{Top Angle}

Defines the vertical angle of the 3D view. The value must be an integer between 0 and 30 .

\section*{Side Angle}

Defines the side angle of the 3D view. The value must be an integer between 0 and 45 .

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:

\section*{Solid Color}

Sets all colors in the color map to solid colors.

\section*{Dark Gradient}

Sets all colors in the color map to a one-color gradient going towards a darker tone.

\section*{Light Gradient}

Sets all colors in the color map to a one-color gradient going towards a lighter tone.

\section*{Glossy}

Gives a glossy look to all bars.
Plot Area Background This control can be used to change the appearance of the plot area backStyle ground. This setting is only available for charts with a plot area. The following options are available:

\section*{Frame}

A frame is drawn around the plot area.

\section*{Shadow}

This option gives a shadow effect on the plot area background.

\section*{Minimal}

This setting removes the plot area background.
Preview Offers a preview of the basic visual properties of the chart.

\subsection*{50.9 Chart Properties: Presentation (Bar- Line- Combo-Radar- Mekko Chart)}


\section*{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)

Cluster Distance ( \(\mathbf{0}-\mathbf{8}\) ) Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.

\section*{Allow Thin Bars}

Show All Bars
Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.

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.

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 609) page.
\[
\begin{array}{ll}
\text { Max Values Shown } & \begin{array}{l}
\text { In this box you can specify an upper limit for the number of data points to } \\
\text { show values for in the chart. If no limit is specified, values will be shown for } \\
\text { all data points, which may affect the readability of the chart. }
\end{array} \\
\text { Vertical } & \begin{array}{l}
\text { Shows the values vertically. }
\end{array} \\
\text { Plot Values Inside Seg- } & \begin{array}{l}
\text { Marking this check box will plot values on data points inside the segments } \\
\text { instead of on top of them. }
\end{array}
\end{array}
\]

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 \(\mathrm{mm}, \mathrm{cm}\), inches ( \("\) ", inch), pixels ( \(\mathrm{px}, \mathrm{pxl}\), pixel), points (pt, pts, point) or docunits (du, docunit).

Symbol Size Determines the size of symbols, if a symbol representation is specified.
Trendline Width
Use Full Symbol Set

Semi-transparent
Pop-up Labels

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.

\section*{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 609), zero values will appear as text above the data points. In other cases zero values will be suppressed.

\section*{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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Limit Legend (Char- \(\quad\)\begin{tabular}{l} 
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 fol- \\
acters)
\end{tabular}\(\quad\)\begin{tabular}{l} 
lowed by ... in the chart.
\end{tabular}
\end{tabular}

In the Chart Scrolling group you can make settings for scrolling in the chart.
\begin{tabular}{ll} 
Enable X-Axis Scroll- & \begin{tabular}{l} 
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 dis- \\
played. The number of values shown at any one time will be the number set \\
under When Number of Items Exceeds.
\end{tabular} \\
Reversed & Checking the box displays the values in reversed order.
\end{tabular}

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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where you can create a new ref- \\
erence line in the chart.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Text in Chart group is used for adding free-floating text to the chart.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog, where you can create a new chart \\
text.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}
\end{tabular}

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 592).

\section*{Legend Settings}


The Legend Settings dialog
The layout of the chart legend is controlled by the various settings of this dialog.
\begin{tabular}{ll} 
Legend Style & Sets the basic style for the legend. Choose between several styles. \\
Background Color & \begin{tabular}{l} 
Sets the color of the legend background. The color can be defined as a solid \\
color or as a gradient via the Color Area (page 412) dialog that opens when \\
clicking the button.
\end{tabular} \\
Vertical Alignment & \begin{tabular}{l} 
Specifies how the legend is positioned in relation to the plot area, when it \\
needs less vertical space than the plot area.
\end{tabular} \\
Font & \begin{tabular}{l} 
Opens the standard Font (page 475) dialog where a font for the legend can be \\
specified.
\end{tabular} \\
Line Spacing & \begin{tabular}{l} 
Specifies the distance between items in the legend.
\end{tabular} \\
Reverse Order & \begin{tabular}{l} 
Reverses the sort order of the legend.
\end{tabular} \\
Multiline & \begin{tabular}{l} 
Sets the options for multi line legend items: \\
Wrap Text \\
Wraps the text of the legend items in two or more lines. \\
Cell Height (Lines)
\end{tabular} \\
If the Wrap Text option is enabled, this setting specifies how many lines that \\
should be used for each item.
\end{tabular}

\section*{Reference Lines}


\section*{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 origins.

Label

\section*{Show Label in Chart}

\section*{Location}

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.

Enable this setting if the label should appear next to the reference line.
Sets from which axis the reference line should originate:

\section*{Continuous \(\mathbf{X}\)}

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 628))

\section*{Primary Y}

The reference line originates from the primary \(y\)-axis (left/bottom).

\section*{Secondary Y}

The reference line originates from the secondary y-axis (right/top).

\section*{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

Show

Defines the layout of the reference line:

\section*{Weight}

Specifies the weight of the reference line. The value can be specified in mm, cm , inches ( \("\), inch), pixels ( \(\mathrm{px}, \mathrm{pxl}\), pixel), points ( pt , pts, point) or docunits (du, docunit).

\section*{Color}

Sets the color for the reference line.
Style
Specifies the style for the reference line, for example, continuous, dashed or dotted.

Specifies the condition for displaying the reference line.

\section*{Always}

The reference line will always be displayed.

\section*{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.

\section*{Chart Text}


The Chart Text dialog
\begin{tabular}{ll} 
Text & \begin{tabular}{l} 
Enter a text that should be displayed in the chart. The text entered can also be \\
defined as Calculated Formula (page 851) for dynamic update. Click the ... \\
button to open the Edit Expression dialog for easier editing of long formulas \\
or for typing multi line text.
\end{tabular} \\
Font & \begin{tabular}{l} 
Opens the standard Font (page 475) dialog where a font for the text can be \\
specified.
\end{tabular} \\
On Top & Forces the text to the foreground when the chart is drawn.
\end{tabular}

\section*{Background}

Angle (degrees)

Alignment

Defines the background of the text.

\section*{Transparent}

With this option, only the text itself will be visible. Any sheet object covered by the text will be fully visible.

\section*{Fixed}

This option lets you pick a background color by clicking the Color button to the right of the radio button.

\section*{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 361). 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.

Specifies the angle for the text. 0 to 360 degrees is allowed, the default value is 0 .

\subsection*{50.10 Chart Properties: Axes (Bar- Line- Combo- RadarMekko Chart)}


\section*{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 \(y\)-axes 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:

\section*{Expression Axes}

Expressions The available expressions displayed here are defined in the Chart Properties: Expressions (page 609) 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.

\section*{Log Scale}

Forced 0

Hide Axis
Show Grid / Show
Minor Grid

A logarithmic scale can be used, provided that all data points in the chart have positive values ( \(>0\) ).

The x -axis will cross at \(\mathrm{y}=0\). This option is not available when a logarithmic axis is used.

Hides the axis for the selected expression.
The scale ticks of the \(y\)-axis that will be used for showing horizontal (y-axes Position is set to Left and/or Right) and vertical (y-axes Position is set to Top and/or Bottom) grid lines.

Scale: \(\quad\) 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 \(\quad\) Check this alternative to set a fixed maximum value for the \(y\)-axis.
Max

Static Step Check this alternative to set a fixed interval between scale ticks for the yaxis.

Values entered in the Scale group may be specified as a Calculated Formula (page 851). Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

\section*{Position}

Split Axis

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.

Splitting the \(y\)-axis into two parts creates the impression of two charts sharing a common x-axis. The Primary \(\boldsymbol{n} \%\) setting defines the percentage of the available axis length that will be used for the primary part of the axis.

\section*{Dimension Axis}

Primary Dimension
Labels
Secondary Dimension
Labels
Continuous
Forced 0

Set the display of labels for main dimension as either horizontal, diagonal or vertical text.

Display labels for secondary dimension as either horizontal, diagonal or vertical text.

Scale axes as continuous numeric (linear).
The y -axis will cross at \(\mathrm{x}=0\).
\begin{tabular}{ll} 
Hide Axis & The x-axis will not be shown. \\
Show Grid & \begin{tabular}{l} 
The tick marks of the x-axis will be extended into grid lines. The Grid layout \\
can be modified via the Grid Style and Grid Color settings (See end of page).
\end{tabular} \\
Show Minor Grid & This alternative enables a subdivision of the grid. \\
Stagger Labels & \begin{tabular}{l} 
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 stag- \\
gered from left to right. If you mark the check box Reverse Stagger, the \\
stagger will be reversed to right to left.
\end{tabular}
\end{tabular}

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: \(\quad\) 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 \(\quad\) Check this alternative to set a fixed maximum value for the x -axis.
Max

Static Step Check this alternative to set a fixed interval between scale ticks for the x axis.

Values entered in the Scale group may be specified as a Calculated Formula (page 851). Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

\section*{Backcast}

\section*{Forecast}

Grid Style

Grid Color
Synchronize Zero Level for ExpressionAxes

This function operates on trend lines. Enter into the text edit box how far back you would like to estimate the trend line. See Chart Properties: Expressions (page 609). The backcast portion of trend lines is displayed as dotted.

Enter into the text edit box how far you wish to predict the trend line. See Chart Properties: Expressions (page 609). The forecast portion of trend lines is displayed as dotted.

With Show Grid checked, you can choose from the available grid styles in the drop-down list.

This alternative lets you pick a color for the grid.
When two y-axes are displayed, this setting is used for synchronizing the zero level of them.

\subsection*{50.11 Chart Properties: Colors}


\section*{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 412) 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 632) dialog where color maps can be set and retrieved on sheet, document, user and QlikView default level.

Multicolored

\section*{Persistent Colors}

Repeat Last Color

Unchecking this option results in all bars having the same color.
Checking this option locks the color map so that each value has a color permanently assigned to it.

Checking this option assigns the 18th color in the color map to any value after the 18 th value. If unchecked, colors are just repeated in succession from 1 to 18.

\section*{Use Patterns Instead of Colors}

\section*{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.
\begin{tabular}{ll} 
Color & \begin{tabular}{l} 
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 412) dialog. \\
Tip! The Background Color setting can be combined with the Image and/or \\
Plot Area Only options below.
\end{tabular} \\
Background & \begin{tabular}{l} 
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.
\end{tabular} \\
Plot Area & \begin{tabular}{l} 
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.
\end{tabular} \\
Image & \begin{tabular}{l} 
Select this alternative and click the Image button to open the Select Image \\
dialog, where it is possible to import a background picture.
\end{tabular} \\
Limit the imported picture to the Plot Area Only by checking this alternative.
\end{tabular}

In the Plot Area Border group you can assign Width and Color for the rectangle surrounding the plot area.

\section*{Advanced Color Map}


The Advanced Color Map dialog

In the this dialog, color settings for several different levels can be retrieved, set or cleared.
\begin{tabular}{ll} 
Current Color Map & \begin{tabular}{l} 
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 \\
Retrieve button for Sheet Default, Document Default, User Default or Qlik- \\
View Default.
\end{tabular} \\
Sheet Default & \begin{tabular}{l} 
Sets, retrieves or clears the default color map for the current sheet. Press \\
Retrieve to retrieve the current sheet default color map (the option is only \\
available if a sheet default is available). Press Update to apply the current \\
color map as sheet default. Press Remove to clear the default for the current \\
sheet (the option is only available if a sheet default is available).
\end{tabular} \\
Document Default & \begin{tabular}{l} 
Sets, retrieves or clears the default color map for the current document. Press \\
Retrieve to retrieve the current document default color map (the option is only \\
available if a document default is available). Press Update to apply the current \\
color map as document default. Press Remove to clear the default for the cur- \\
rent document (the option is only available if a document default is available).
\end{tabular} \\
User Default & \begin{tabular}{l} 
Sets, retrieves or clears the default color map for the current user. Press \\
Retrieve to retrieve the current user default color map (the option is only avail- \\
able if a user default is available). Press Update to apply the current color map \\
as user default. Press Remove to clear the default for the current user (the \\
option is only available if a user default is available).
\end{tabular} \\
QlikView Default & \begin{tabular}{l} 
Press Retrieve to retrieve the QlikView default color map. This default cannot
\end{tabular} \\
be changed.
\end{tabular}

\subsection*{50.12 Chart Properties: Number}


\section*{Chart Properties, Number}

This property page applies to the active chart and contains the following controls for formatting values:
Expression Default Shows numeric values using the number format provided by the expression.
\begin{tabular}{ll} 
Number & \begin{tabular}{l} 
Shows numeric values with the number of digits set in the Precision spinner \\
box.
\end{tabular} \\
Integer & Shows numeric values as integers. \\
Fixed to & \begin{tabular}{l} 
Shows numeric values as decimal values with the number of decimal digits set \\
in the Decimals spinner box.
\end{tabular} \\
Money & \begin{tabular}{l} 
Shows numeric values in the format shown in the Preview text box. The \\
default format is the Windows Currency setting.
\end{tabular} \\
Date & \begin{tabular}{l} 
Shows values that can be interpreted as dates in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular} \\
Time & \begin{tabular}{l} 
Shows values that can be interpreted as time in the format set in the Format \\
Pattern edit box. An example of this format is shown in the Preview text box.
\end{tabular}
\end{tabular}

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.

\section*{Interval}

Shows time as sequential time increment (e.g. format \(=m m\) 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, 1000000 and 1000000000 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.

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) 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 747) from the Tools menu.
By a right-click on the line chart the Line Chart: Object Menu (page 638) 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 595).

\subsection*{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.

\subsection*{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:
Properties... Opens the Properties dialog where the parameters defining the chart can be set.

Detach 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.

\section*{Attach}

\section*{Set Reference}

\section*{Clear Reference}

Clone Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.

Order This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) checkbox is marked. It contains four commands to set the layout layer of the sheet objects. Valid layer numbers are -128 to 127 .

\section*{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.

\section*{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.

\section*{Bring Forward}

Increases the layout layer of the sheet object by one. Maximum value is 127 .

\section*{Send Backward}

Decreases the layout layer of the sheet object by one. Minimum value is -128 .
\begin{tabular}{ll} 
Clear All Selections & Clears all selections in the dimensions and expressions of the chart. \\
Print... & \begin{tabular}{l} 
Opens the Print: General (page 99) dialog from which you can print the \\
chart.
\end{tabular} \\
Print as PDF... & \begin{tabular}{l} 
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 Qlik- \\
View's homepage and installed separately.
\end{tabular} \\
Send Values to Excel \(\quad\)\begin{tabular}{l} 
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
\end{tabular} \\
Microsoft Excel 97 or later must be installed on the computer. \\
Copy to Clipboard & \begin{tabular}{l} 
Opens a dialog for saving an image of the chart to file. The image can be \\
saved as png, jpg, bmp or gif.
\end{tabular} \\
This cascade menu contains the various copy options for the sheet object. \\
Remove & \begin{tabular}{l} 
Values \\
Copies the values to the clipboard in the form of a table.
\end{tabular} \\
Image \\
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
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{51.9 Chart Properties: Presentation (Bar- Line- Combo-Radar- Mekko Chart)}


\section*{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)

Cluster Distance ( \(\mathbf{0}-\mathbf{8}\) ) Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.

\section*{Allow Thin Bars}

Show All Bars
Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.

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.

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 609) page.
\begin{tabular}{ll} 
Max Values Shown & \begin{tabular}{l} 
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.
\end{tabular} \\
Vertical & Shows the values vertically. \\
Plot Values Inside Seg- & \begin{tabular}{l} 
Marking this check box will plot values on data points inside the segments \\
ments
\end{tabular} \\
instead of on top of them.
\end{tabular}

In the Error Bars group display options for any error bars used in the chart is determined
\begin{tabular}{ll} 
Width & Specifies the width of error bars. \\
Thickness & Specifies the thickness of error bars \\
Color & Sets a color for error bars.
\end{tabular}

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.
\begin{tabular}{ll} 
Line Width & \begin{tabular}{l} 
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).
\end{tabular} \\
Symbol Size & \begin{tabular}{l} 
Determines the size of symbols, if a symbol representation is specified.
\end{tabular} \\
Trendline Width & This setting determines the width of trendlines.
\end{tabular} Use Full Symbol Set \(\quad\)\begin{tabular}{l} 
This alternative makes more symbol representation available (rings, triangles \\
etc.)
\end{tabular}
\begin{tabular}{ll} 
Semi-transparent & Check this option if you want filled lines to be drawn semi-transparent. \\
Pop-up Labels & \begin{tabular}{l} 
Check this option to display the corresponding dimension value in a pop-up \\
window when the mouse pointer touches a value.
\end{tabular} \\
Highlight & \begin{tabular}{l} 
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 sev- \\
eral overlapping values.
\end{tabular} \\
Suppress Zero-Values \(\quad\)\begin{tabular}{l} 
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 \\
609), zero values will appear as text above the data points. In other cases zero \\
values will be suppressed.
\end{tabular} \\
Suppress Missing & \begin{tabular}{l} 
If this check box is marked, all combinations of the dimension fields asso- \\
ciated 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 use- \\
ful only in special cases, e.g. if you want to count null values in a chart.
\end{tabular}
\end{tabular}

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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Limit Legend (Char- & \begin{tabular}{l} 
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 fol- \\
acters) \\
lowed by ... in the chart.
\end{tabular}
\end{tabular}

In the Chart Scrolling group you can make settings for scrolling in the chart.
\begin{tabular}{ll} 
Enable X-Axis Scroll- & \begin{tabular}{l} 
Enable this check box to show a scroll control in place of the X-axis. The \\
bar \\
scroll bar can be used to scroll the selection of the X-axis values that are dis- \\
played. The number of values shown at any one time will be the number set \\
under When Number of Items Exceeds.
\end{tabular} \\
Reversed & Checking the box displays the values in reversed order.
\end{tabular}

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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where you can create a new ref- \\
erence line in the chart.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Text in Chart group is used for adding free-floating text to the chart.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog, where you can create a new chart \\
text.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}
\end{tabular}

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 592).

\subsection*{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 628).

\subsection*{51.11 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) 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 645) 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 595).

\subsection*{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:
Properties... Opens the Properties dialog where the parameters defining the chart can be set.

\section*{Detach}

Attach

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.

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone \begin{tabular}{l} 
Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58 ) is activated or when the Always Show Design Menu
\end{tabular} \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\quad\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{52.8 Chart Properties: Presentation (Bar- Line- Combo-Radar- Mekko Chart)}


\section*{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.

\begin{abstract}
Cluster Distance ( \(\mathbf{0}-\mathbf{8}\) ) 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.
\end{abstract}

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 609) page.
\begin{tabular}{ll} 
Max Values Shown & \begin{tabular}{l} 
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.
\end{tabular} \\
Vertical & \begin{tabular}{l} 
Shows the values vertically.
\end{tabular} \\
\begin{tabular}{ll} 
Plot Values Inside Seg- \\
ments
\end{tabular} & \begin{tabular}{l} 
Marking this check box will plot values on data points inside the segments \\
instead of on top of them.
\end{tabular}
\end{tabular}

In the Error Bars group display options for any error bars used in the chart is determined
\begin{tabular}{ll} 
Width & Specifies the width of error bars. \\
Thickness & Specifies the thickness of error bars. \\
Color & Sets a color for error bars.
\end{tabular}

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.
\begin{tabular}{ll} 
Line Width & \begin{tabular}{l} 
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).
\end{tabular} \\
Symbol Size & \begin{tabular}{l} 
Determines the size of symbols, if a symbol representation is specified.
\end{tabular} \\
Trendline Width & \begin{tabular}{l} 
This setting determines the width of trendlines.
\end{tabular} \\
Use Full Symbol Set & \begin{tabular}{l} 
This alternative makes more symbol representation available (rings, triangles \\
etc.)
\end{tabular} \\
Semi-transparent & Check this option if you want filled lines to be drawn semi-transparent. \\
Pop-up Labels & \begin{tabular}{l} 
Check this option to display the corresponding dimension value in a pop-up \\
window when the mouse pointer touches a value.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Highlight & \begin{tabular}{l} 
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 sev- \\
eral overlapping values.
\end{tabular} \\
Suppress Zero-Values \(\quad\)\begin{tabular}{l} 
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 \\
609), zero values will appear as text above the data points. In other cases zero \\
values will be suppressed.
\end{tabular} \\
Suppress Missing & \begin{tabular}{l} 
If this check box is marked, all combinations of the dimension fields asso- \\
ciated 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 use- \\
ful only in special cases, e.g. if you want to count null values in a chart.
\end{tabular}
\end{tabular}

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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Limit Legend (Char- \(\quad\)\begin{tabular}{l} 
Enable this check box to limit the length of the dimension value strings that \\
ace displayed on axes and in the chart legend. Truncated values will be fol- \\
lowed by ... in the chart.
\end{tabular}
\end{tabular}

In the Chart Scrolling group you can make settings for scrolling in the chart.
Enable X-Axis Scroll- Enable this check box to show a scroll control in place of the X-axis. The bar

\section*{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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where you can create a new ref- \\
erence line in the chart.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Text in Chart group is used for adding free-floating text to the chart.
Add
Opens the Chart Text (page 627) 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 627) dialog.

\section*{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 592).

\subsection*{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 628).

\subsection*{52.10 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) 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 653) 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 595).

\subsection*{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:
Properties... Opens the Properties dialog where the parameters defining the chart can be set.

\section*{Detach}

Attach

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.

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone \begin{tabular}{l} 
Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58 ) is activated or when the Always Show Design Menu
\end{tabular} \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\square\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking \({ }^{\square}\) 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{53.8 Chart Properties: Presentation (Bar- Line- Combo-Radar- Mekko Chart)}


\section*{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.

\begin{abstract}
Cluster Distance ( \(\mathbf{0}-\mathbf{8}\) ) 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.
\end{abstract}

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 609) page.
\begin{tabular}{ll} 
Max Values Shown & \begin{tabular}{l} 
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.
\end{tabular} \\
Vertical & \begin{tabular}{l} 
Shows the values vertically.
\end{tabular} \\
\begin{tabular}{ll} 
Plot Values Inside Seg- \\
ments
\end{tabular} & \begin{tabular}{l} 
Marking this check box will plot values on data points inside the segments \\
instead of on top of them.
\end{tabular}
\end{tabular}

In the Error Bars group display options for any error bars used in the chart is determined
\begin{tabular}{ll} 
Width & Specifies the width of error bars. \\
Thickness & Specifies the thickness of error bars. \\
Color & Sets a color for error bars.
\end{tabular}

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.
\begin{tabular}{ll} 
Line Width & \begin{tabular}{l} 
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).
\end{tabular} \\
Symbol Size & \begin{tabular}{l} 
Determines the size of symbols, if a symbol representation is specified.
\end{tabular} \\
Trendline Width & \begin{tabular}{l} 
This setting determines the width of trendlines.
\end{tabular} \\
Use Full Symbol Set & \begin{tabular}{l} 
This alternative makes more symbol representation available (rings, triangles \\
etc.)
\end{tabular} \\
Semi-transparent & Check this option if you want filled lines to be drawn semi-transparent. \\
Pop-up Labels & \begin{tabular}{l} 
Check this option to display the corresponding dimension value in a pop-up \\
window when the mouse pointer touches a value.
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Highlight & \begin{tabular}{l} 
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 sev- \\
eral overlapping values.
\end{tabular} \\
Suppress Zero-Values \(\quad\)\begin{tabular}{l} 
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 \\
609), zero values will appear as text above the data points. In other cases zero \\
values will be suppressed.
\end{tabular} \\
Suppress Missing & \begin{tabular}{l} 
If this check box is marked, all combinations of the dimension fields asso- \\
ciated 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 use- \\
ful only in special cases, e.g. if you want to count null values in a chart.
\end{tabular}
\end{tabular}

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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Limit Legend (Char- \(\quad\)\begin{tabular}{l} 
Enable this check box to limit the length of the dimension value strings that \\
ace displayed on axes and in the chart legend. Truncated values will be fol- \\
lowed by ... in the chart.
\end{tabular}
\end{tabular}

In the Chart Scrolling group you can make settings for scrolling in the chart.
Enable X-Axis Scroll- Enable this check box to show a scroll control in place of the X-axis. The bar

\section*{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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where you can create a new ref- \\
erence line in the chart.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Text in Chart group is used for adding free-floating text to the chart.
Add
Opens the Chart Text (page 627) 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 627) dialog.

\section*{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 592).

\subsection*{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 628).

\subsection*{53.10 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) 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 661) 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 595).

\subsection*{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:
Properties... Opens the Properties dialog where the parameters defining the chart can be set.

Detach 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.

Attach Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone \begin{tabular}{l} 
Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58 ) is activated or when the Always Show Design Menu
\end{tabular} \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\quad\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{54.5 Chart Properties: Expressions (Scatter Chart)}


\section*{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 \(\mathbf{X}\) 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 851). Click the ... button to open the Edit Expression dialog for easier editing of long formulas.
In the \(\mathbf{Z}\) 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.

\subsection*{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 618).

\subsection*{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 619).

\subsection*{54.8 Chart Properties: Presentation (Scatter- Grid Chart)}


\section*{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:
\begin{tabular}{ll} 
Only Lines & Data is represented by lines between data points. \\
Only Symbols & Data is represented by symbols, drawn at the data points. \\
\begin{tabular}{l} 
Both Lines and Sym- \\
bols
\end{tabular} & A combination of the alternatives above. \\
Line Width & \begin{tabular}{l} 
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).
\end{tabular} \\
Trendline Width & \begin{tabular}{l} 
This setting determines the width of trend lines.
\end{tabular} \\
Symbol Size & Determines the size of symbols, if a symbol representation is specified. \\
Max Bubble Size & Sets the size of the largest bubble in the chart when bubble style is used.
\end{tabular}
\begin{tabular}{ll} 
Show Arrows & \begin{tabular}{l} 
Set display options for arrows on connecting lines (See the Representation \\
group, above!). Arrows will be directed between scatter points in the sequen- \\
tial order defined by the sort order of the second field dimension. Check this \\
alternative to have arrow heads displayed on the connecting lines.
\end{tabular} \\
Arrow Size & The size of the arrow head can be edited here. \\
Style & Select from several styles from the drop-down list.
\end{tabular}

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.
\begin{tabular}{ll} 
Max Labels Shown & \begin{tabular}{l} 
Limits the number of plotted labels. Setting this number too high may degrade \\
chart clarity.
\end{tabular} \\
Labels on Datapoints & \begin{tabular}{l} 
Check this alternative to have labels displayed.
\end{tabular} \\
Pop-up Labels & \begin{tabular}{l} 
Check this alternative to display the corresponding dimension value in a pop- \\
up window when pointing in the chart.
\end{tabular} \\
Highlight & \begin{tabular}{l} 
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 sev- \\
eral overlapping values.
\end{tabular} \\
Horizontal Position & \begin{tabular}{l} 
Select the horizontal orientation from the drop-down list: Left, Centered or \\
Right.
\end{tabular} \\
Vertical Position & \begin{tabular}{l} 
Select the vertical orientation from the drop-down list: Above, Centered or \\
Below.
\end{tabular}
\end{tabular}

The check boxes Suppress Zero-Values and Suppress Missing eliminate dimensions that are empty or contain only zeros from the table.

Show X Label The label of x-dimension specified in the \(\mathbf{X}\) group of Chart Properties:
Expressions (Scatter Chart) (page 664) page is drawn at the end of the x -axis.
Show Y Label The label of y-dimension specified in the \(\mathbf{Y}\) group of Chart Properties:
Expressions (Scatter Chart) (page 664) 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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Limit Legend (Char- & \begin{tabular}{l} 
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 fol- \\
lowed by ... in the chart.
\end{tabular}
\end{tabular}

In the Chart Scrolling group you can make settings for scrolling in the chart.

Enable X-Axis Scroll- Enable this check box to show a scroll control in place of the X-axis. The bar 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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where a new reference line in \\
the chart can be created.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Texts in Chart group is used for adding free-floating text to the chart.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog where a new chart text in the chart \\
can be created.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}
\end{tabular}

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 592).
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 origins.
\begin{tabular}{|c|c|}
\hline 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. \\
\hline Show Label in Chart & Enable this setting if the label should appear next to the reference line. \\
\hline Location & Sets from which axis the reference line should originate, choose between \(\mathbf{X}\) axis, Y -axis and X -axis and Y -axis \\
\hline 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. \\
\hline Line Formatting & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Show & \begin{tabular}{l}
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.
\end{tabular} \\
\hline
\end{tabular}

\subsection*{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:

\section*{Forced 0}

Show Grid / Show Minor Grid

Log Scale
Label Along Axis

\section*{Hide Axis}

The y -axis will cross at \(\mathrm{x}=0\).
The scale ticks of the \(x\)-axis will be extended into grid lines that can be further formatted by the Grid Style and Grid Color settings.

A logarithmic scale will be used.
Changes the position of the x -axis label.
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 \(\quad\) Check this alternative to set a fixed maximum value for the x -axis.
Max

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 851). Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

\section*{Y Axis:}
Forced \(0 \quad\) The x -axis will cross at \(\mathrm{y}=0\)

\section*{Show Grid / Show}

Minor Grid
Log Scale
Hide Axis
Label Along Axis

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.

A logarithmic scale will be used.
The \(y\)-axis will not be shown.
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.
\begin{tabular}{lll} 
Scale: & \begin{tabular}{l} 
Static \\
Min
\end{tabular} & \begin{tabular}{l} 
The scale of the y-axis will not change with document state. Check this alter- \\
native to set a fixed minimum value for the \(y\)-axis in the edit box.
\end{tabular} \\
\begin{tabular}{l} 
Static \\
Max
\end{tabular} & Check this alternative to set a fixed maximum value for the y-axis.
\end{tabular}\(\quad\)\begin{tabular}{l} 
Static \\
Step
\end{tabular} \begin{tabular}{l} 
Check this alternative to set a fixed interval between scale ticks for the y- \\
axis.
\end{tabular}

Values entered in the Scale group may be specified as a Calculated Formula (page 851). Click the ... button to open the Edit Expression dialog for easier editing of long formulas.

\section*{Grid Style}

Grid Color This alternative lets you pick a color for the grid.

\subsection*{54.10 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{55 Grid Chart}


\section*{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 593) 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 673) 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 595).

\subsection*{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:
Properties... Opens the Properties dialog where the parameters defining the chart can be set.

\section*{Detach}

\section*{Attach}

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.

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone \begin{tabular}{l} 
Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58 ) is activated or when the Always Show Design Menu
\end{tabular} \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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.
\end{tabular}
\begin{tabular}{ll} 
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image
\end{tabular} \\
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 User Preferences: Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the lay- \\
out or in another document opened within the current instance of QlikView. \\
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.
\end{tabular}

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{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:
\begin{tabular}{ll} 
Only Lines & Data is represented by lines between data points. \\
Only Symbols & Data is represented by symbols, drawn at the data points. \\
\begin{tabular}{l} 
Both Lines and Sym- \\
bols
\end{tabular} & A combination of the alternatives above. \\
Line Width & \begin{tabular}{l} 
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).
\end{tabular}
\end{tabular}
\begin{tabular}{ll} 
Trendline Width & This setting determines the width of trend lines. \\
Symbol Size & Determines the size of symbols, if a symbol representation is specified. \\
Max Bubble Size & Sets the size of the largest bubble in the chart when bubble style is used. \\
Show Arrows & \begin{tabular}{l} 
Set display options for arrows on connecting lines (See the Representation \\
group, above!). Arrows will be directed between scatter points in the sequen- \\
tial order defined by the sort order of the second field dimension. Check this \\
alternative to have arrow heads displayed on the connecting lines.
\end{tabular} \\
Arrow Size & \begin{tabular}{l} 
The size of the arrow head can be edited here.
\end{tabular} \\
Style & Select from several styles from the drop-down list.
\end{tabular}

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.
\begin{tabular}{ll} 
Max Labels Shown & \begin{tabular}{l} 
Limits the number of plotted labels. Setting this number too high may degrade \\
chart clarity.
\end{tabular} \\
Labels on Datapoints & \begin{tabular}{l} 
Check this alternative to have labels displayed.
\end{tabular} \\
Pop-up Labels & \begin{tabular}{l} 
Check this alternative to display the corresponding dimension value in a pop- \\
up window when pointing in the chart.
\end{tabular} \\
Highlight & \begin{tabular}{l} 
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 sev- \\
eral overlapping values.
\end{tabular} \\
Horizontal Position & \begin{tabular}{l} 
Select the horizontal orientation from the drop-down list: Left, Centered or \\
Right.
\end{tabular} \\
Vertical Position & \begin{tabular}{l} 
Select the vertical orientation from the drop-down list: Above, Centered or \\
Below.
\end{tabular}
\end{tabular}

The check boxes Suppress Zero-Values and Suppress Missing eliminate dimensions that are empty or contain only zeros from the table.
\begin{tabular}{ll} 
Show X Label & \begin{tabular}{l} 
The label of x-dimension specified in the \(\mathbf{X}\) group of Chart Properties: \\
Expressions (Scatter Chart) (page 664) page is drawn at the end of the x-axis.
\end{tabular} \\
Show Y Label & The label of y-dimension specified in the \(\mathbf{Y}\) group of Chart Properties: \\
& Expressions (Scatter Chart) (page 664) page is drawn at the end of the y-axis.
\end{tabular}

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.

\section*{Show Legend}

Check this alternative to include a legend in the chart (checked by default). It is possible to change the Legend Settings (page 625) 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.

\section*{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- Enable this check box to show a scroll control in place of the X-axis. The bar 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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where a new reference line in \\
the chart can be created.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Texts in Chart group is used for adding free-floating text to the chart.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog where a new chart text in the chart \\
can be created.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}
\end{tabular}

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 592).

\section*{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 origins.

Label

\section*{Show Label in Chart}

\section*{Location}

\section*{Definition}

\section*{Line Formatting}

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.

Enable this setting if the label should appear next to the reference line.
Sets from which axis the reference line should originate, choose between \(\mathbf{X}\) axis, Y -axis and X -axis and Y -axis

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.

Defines the layout of the reference line:

\section*{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).

\section*{Color}

Sets the color for the reference line.

\section*{Style}

Specifies the style for the reference line, for example, continuous, dashed or dotted.

Show Specifies the condition for displaying the reference line.

\section*{Always}

The reference line will always be displayed.

\section*{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.

\subsection*{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 628).

\subsection*{55.9 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) 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 747) from the Tools menu.
By a right-click on the pie chart the Pie Chart: Object Menu (page 681) 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 595).

\subsection*{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:

Properties...

\section*{Detach}

\section*{Attach}

Opens the Properties dialog where the parameters defining the chart can be set.

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.

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone \begin{tabular}{l} 
Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
View Menu (page 58 ) is activated or when the Always Show Design Menu
\end{tabular} \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\quad\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{56.7 Chart Properties: Presentation (Pie Chart)}


\section*{Chart Properties, Presentation (Pie Chart)}

In this dialog you specify settings that determine how the pie-slices are displayed.
The default values are:

\section*{Pop-up Labels}

Suppress Zero-Values

Suppress Missing

Enable this setting to display the corresponding dimension value in a pop-up window when the mouse pointer touches the value.

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.

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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Show Numbers in Leg- & If enabled, a legend showing numerical values will be included in the chart. \\
end
\end{tabular}\(\quad\)\begin{tabular}{ll} 
Limit Legend (Char- & \begin{tabular}{l} 
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 fol- \\
acters)
\end{tabular} \\
lowed by ... in the chart.
\end{tabular}

The Texts in Chart group is used for adding free-floating text to the chart.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog, where you can create a new chart \\
text.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}

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 592).

\subsection*{56.8 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 681). 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 593) page.
By a right-click on the funnel chart the Funnel Chart: Object Menu (page 687) 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 595).

\subsection*{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:

Properties...

\section*{Detach}

\section*{Attach}

Opens the Properties dialog where the parameters defining the chart can be set.

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.

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone \begin{tabular}{l} 
Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular} \\
Order & \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the
\end{tabular} \\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out layer of the sheet objects. Valid layer numbers are -128 to 127. \\
Bring to Front
\end{tabular}
\begin{tabular}{|c|c|}
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\quad\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{57.8 Chart Properties: Presentation (Funnel Chart)}


\section*{Chart Properties, Presentation (Funnel Chart)}

By changing the settings on this page, you modify the way the funnel chart is presented on the screen.
Pop-up Labels 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.

Tip Width (\%) Defines the width of the funnel tip as a percentage of the funnel mouth width.
\begin{tabular}{ll} 
Reversed Orientation & \begin{tabular}{l} 
Enable this check box if you want to make the funnel point to the left or \\
upwards
\end{tabular} \\
Data Proportionality \(\quad\) & This setting controls the degree to which the segments are proportional to the \\
underlying data. \\
& Equal Segment Heights \\
& No proportionality to data. Each segment is drawn with the same height \\
(width for horizontal funnels) as all others. Typically only makes sense with \\
& numbers displayed in the chart. \\
& Segment Height Proportional to Data \\
& Each segment is drawn with the a height (width for horizontal funnels) pro- \\
portional to underlying data. \\
& Segment Area Proportional to Data \\
& Each segment is drawn with the total area proportional to underlying data. \\
Show Legend & In the Show Legend group you can control the display of dimension data \\
& labels in the chart. Mark the check box to display data labels. Data labels are \\
shown only for the current top level of the chart. \\
& Settings... \\
& Opens the Legend Settings (page 625) dialog where the legend settings can \\
& be modified. \\
& Show Numbers in Legend \\
& Enable this check box to show values next to the dimension data labels. \\
& 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 fol- \\
lowed by ... in the chart.
\end{tabular}

\subsection*{57.9 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) 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 693) 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 595).

\subsection*{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:

Properties... Opens the Properties dialog where the parameters defining the chart can be set.

Detach 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.

Attach

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{ll} 
Set Reference & \begin{tabular}{l} 
By choosing this option you set a chart reference, i.e. a fixed plot of the chart \\
with the current selections. When further selections are made in the document \\
the reference plot remains, dimmed in the background. Chart axes etc will be \\
adjusted to always include the maximum of the background data set and the \\
current data set. The current data set is always plotted on top of the reference \\
plot, i.e. some portions of the reference plot may be obscured by the current \\
data set plot. The way the background is dimmed can be controlled via the \\
Reference Mode setting on the Chart Properties: General 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.
\end{tabular} \\
Clear Reference & \begin{tabular}{l} 
This command is replaced with the Set Reference 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.
\end{tabular} \\
Clone & Makes an identical copy of the chart. If a detached chart is cloned the clone \\
will be attached.
\end{tabular}
\begin{tabular}{ll} 
Copy to Clipboard & \begin{tabular}{l} 
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include \\
or exclude the 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 lay- \\
out or in another document opened within the current instance of QlikView.
\end{tabular} \\
Opens a menu with the following commands for linked objects. \\
Adjust Position of Linked Objects
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 619).

\subsection*{58.7 Chart Properties: Presentation (Block Chart)}


\section*{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 \(\mathbf{3}\) 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.)

\footnotetext{
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.
}

1st Dimension Relative With this option, the percentage of the indicated 1st dimension value relative Total to the total is included in the pop-up.

2nd Dimension Relative Total

3rd Dimension Relative Total

\author{
Value Relative 1st
} Dimension

Value Relative 2nd Dimension

The percentage of the indicated 2 nd dimension value as defined within the indicated 1 st dimension value relative to the total.

As above, but for the 3rd dimension level.

The percentage of the indicated block relative to the total of the indicated 1 st dimension value.

The percentage of the indicated block relative to the total of the indicated 2nd dimension value.

\section*{Show Dimension} Labels

Check this alternative to include labels in the chart (checked by default). It is also possible to change the Font (page 475) settings, by clicking the corresponding button.
If Show Numbers with Labels 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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog, where it is possible to create a new \\
chart text.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}
\end{tabular}

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 592).

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 625) 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.

```

\section*{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 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
\begin{tabular}{ll} 
Font... & Opens the Font dialog for setting a caption font. \\
Color & Opens the Color Area dialog for setting a caption base color. \\
Intermediate Level & \begin{tabular}{l} 
Mark this check box if you want captions drawn on the intermediate dimen- \\
sion level in three-dimensional block charts.
\end{tabular} \\
Captions &
\end{tabular}

\subsection*{58.8 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{59 Gauge Chart}


\section*{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 593) 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 747) from the Tools menu.
By a right-click on the gauge chart the Gauge Chart: Object Menu (page 699) 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 595).

\subsection*{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:

Properties... Opens the Properties dialog where the parameters defining the chart can be set.

Detach 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.

\section*{Attach}

Attaches a detached chart. The chart becomes dynamically linked to the data. The command is available only if the chart is detached.
\begin{tabular}{|c|c|}
\hline Set Reference & By choosing this option you set a chart reference, i.e. a fixed plot of the chart with the current selections. When further selections are made in the document the reference plot remains, dimmed in the background. Chart axes etc will be adjusted to always include the maximum of the background data set and the current data set. The current data set is always plotted on top of the reference plot, i.e. some portions of the reference plot may be obscured by the current data set plot. The way the background is dimmed can be controlled via the Reference Mode setting on the Chart Properties: General 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. \\
\hline Clear Reference & This command is replaced with the Set Reference 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. \\
\hline Clone & Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached. \\
\hline Order & \begin{tabular}{l}
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) 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 .
\end{tabular} \\
\hline Print... & Opens the Print: General (page 99) dialog from which you can print the chart. \\
\hline Print as PDF... & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Send Values to Excel & 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. \\
\hline Export... & Opens a dialog for saving an image of the chart to file. The image can be saved as png, jpg, bmp or gif. \\
\hline
\end{tabular}
\begin{tabular}{|c|c|}
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\square\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking \({ }^{\square}\) 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{59.7 Chart Properties: Presentation (Gauge Chart)}


\section*{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 737).

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 737).
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.
\begin{tabular}{ll} 
Add & Click this button to add a new segment to the gauge. \\
Delete & Click this button to remove the currently selected segment from the gauge. \\
Promote & \begin{tabular}{l} 
Use this button for sorting the segments in the list.
\end{tabular} \\
Label & \begin{tabular}{l} 
Specifies a name for the selected segment. This name is for identification only \\
and is not used when drawing the gauge chart.
\end{tabular} \\
Lower Bound & \begin{tabular}{l} 
Specifies the gauge value where the selected segment starts. If the Autowidth \\
Segments 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 Relative \\
Segment Bounds 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 Min and Max values.
\end{tabular} \\
Color & \begin{tabular}{l} 
Assign a color to the selected segment by clicking the colored button. This \\
opens the Color Area (page 412) dialog.
\end{tabular}
\end{tabular}

In the Indicator group you determine display options for the gauge indicator. This group is not available for LED style gauge charts.
\begin{tabular}{ll} 
Mode & Select one of the indication modes from the drop-down list. \\
Style & Select one of the gauge variants from the drop-down list.
\end{tabular}

Show Scale The number entered for \(n\) Major Units determines the main division of the scale.
The labeling of the scale can be modified by the Show Labels on Every \(n\) Major Unit setting and the Font button.
The scale can be further detailed by the \(n\) Minor Units per Major Unit setting.

In the Circular Gauge Settings group, which is specific to circular style gauge charts, you specify the properties of the gauge shape.

Cylinder Thickness The gauge is normally drawn as a solid circle or circle segment. The greater the number here, the thicker the cylinder. The value which indicates the percentage of the radius to be left unfilled must be between 0 and 99 .

Angle Span The angle in degrees between the Min and Max values in the gauge. Must be a value between 45 and 360 .

Center Angle The angle of the gauge's center value in relation to the clock. Must be a value between 0 and 360.0 indicates the center at the top of the gauge ( 12 o'clock).

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 626) dialog where you can create a new reference line in the chart.
\begin{tabular}{ll} 
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Text in Chart group is used for adding free-floating text to the chart.
Add Opens the Chart Text (page 627) 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 627) 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 592).
\begin{tabular}{ll} 
Autowidth Segments & \begin{tabular}{l} 
While this option is checked (default), the sizing of segments is calculated \\
automatically, based on the gauge's Min value, Max value and the number of \\
segments defined. If the option is deselected, the following options become \\
available: Lower Bound, which is available from the Segments Setup group, \\
Relative Segment Bounds and Logarithmic Scale.
\end{tabular} \\
Relative Segment & \begin{tabular}{l} 
If this option is selected the segment bounds can be entered as numbers \\
between 0 and 1 indicating a fraction of the total range between the Min and \\
Max values.
\end{tabular} \\
Bide Segment Bound- & \begin{tabular}{l} 
If this option is selected no outline will be drawn between the segment bound- \\
aries of circular and linear gauges, which is useful for creating background gra- \\
dients with more than two colors.
\end{tabular} \\
aries & \begin{tabular}{l} 
If this option is selected no outline will be drawn around the gauge.
\end{tabular} \\
Hide Gauge Outlines & \begin{tabular}{l} 
A logarithmic scale can be used, provided that all data points have positive \\
values (>0).
\end{tabular} \\
Pop-up Labels & \begin{tabular}{l} 
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.
\end{tabular}
\end{tabular}

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)

\section*{Digits}

Color

Style

Specifies the number of LED display digits shown.
Assign a color to the selected LED by clicking the colored button. This opens the Color Area (page 412) dialog.

Select one of the LED variants from the drop-down list.

\subsection*{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 532).

\subsection*{59.9 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{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 593) page.
Mekko charts present data using variable width bars. They can display up to three levels of data in a twodimensional chart. Mekko charts are useful in such areas as market analysis.
By a right-click on the chart the Mekko Chart: Object Menu (page 707) 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 595).

\subsection*{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 twodimensional 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:
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.

\section*{Detach}

\section*{Attach}

Clone Makes an identical copy of the chart. If a detached chart is cloned the clone will be attached.
\begin{tabular}{|c|c|}
\hline Order & \begin{tabular}{l}
This cascade menu is only available when the Design Grid command of the View Menu (page 58) is activated or when the Always Show Design Menu Items (page 86) 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 .
\end{tabular} \\
\hline Clear All Selections & Clears all selections in the dimensions and expressions of the chart. \\
\hline Print... & Opens the Print: General (page 99) dialog from which you can print the chart. \\
\hline Print as PDF... & \begin{tabular}{l}
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. 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.
\end{tabular} \\
\hline Send Values to Excel & 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. \\
\hline Export... & Opens a dialog for saving an image of the chart to file. The image can be saved as bmp, jpg, gif or png. \\
\hline Copy to Clipboard & This menu contains the various copy options for the chart. \\
\hline & \begin{tabular}{l}
Values \\
Copies the values to the clipboard in the form of a table. \\
Image \\
Copies an image of the chart object to the clipboard. The image will include or exclude the 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.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline
\end{tabular}
\begin{tabular}{ll} 
Minimize & \begin{tabular}{l} 
Iconizes the object. Clicking \\
same result. This command is available only if minimizing is allowed in the \\
object's Properties dialog on the Caption (page 479) page.
\end{tabular} \\
Maximize & \begin{tabular}{l} 
Enlarges the object to fill the screen. Clicking 回 in the object caption (if \\
shown) produces the same result. This command is available only if max- \\
imizing is allowed in the object's Properties dialog on the Caption (page \\
479) page.
\end{tabular} \\
Restore & \begin{tabular}{l} 
Restores a minimized or maximized object to its previous size and location. \\
Double-clicking the icon of a minimized object or clicking 圆 in the object \\
caption (if shown) of a maximized object produces the same result. This com- \\
mand is available only for minimized or maximized objects.
\end{tabular} \\
Help & \begin{tabular}{l} 
Opens QlikView help.
\end{tabular} \\
Remove & Removes the active sheet object from the sheet.
\end{tabular}

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{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 619).

\subsection*{60.8 Chart Properties: Presentation (Bar- Line- Combo-Radar- Mekko Chart)}


\section*{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)

Cluster Distance ( \(\mathbf{0}-\mathbf{8}\) ) Denotes the distance between grouped values in a clustered bar chart. Values between 0 and 8 are allowed.

\section*{Allow Thin Bars}

\section*{Show All Bars}

Sets the distance between the bars in the cluster. A negative number results in overlapping bars. Values between -6 and 8 are allowed.

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.

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 609) page.
\[
\begin{array}{ll}
\text { Max Values Shown } & \begin{array}{l}
\text { In this box you can specify an upper limit for the number of data points to } \\
\text { show values for in the chart. If no limit is specified, values will be shown for } \\
\text { all data points, which may affect the readability of the chart. }
\end{array} \\
\text { Vertical } & \begin{array}{l}
\text { Shows the values vertically. }
\end{array} \\
\text { Plot Values Inside Seg- } & \begin{array}{l}
\text { Marking this check box will plot values on data points inside the segments } \\
\text { instead of on top of them. }
\end{array}
\end{array}
\]

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 \(\mathrm{mm}, \mathrm{cm}\), inches ( \("\) ", inch), pixels ( \(\mathrm{px}, \mathrm{pxl}\), pixel), points (pt, pts, point) or docunits (du, docunit).

Symbol Size Determines the size of symbols, if a symbol representation is specified.
Trendline Width
Use Full Symbol Set

Semi-transparent
Pop-up Labels

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.

\section*{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 609), zero values will appear as text above the data points. In other cases zero values will be suppressed.

\section*{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.
\begin{tabular}{ll} 
Show Legend & \begin{tabular}{l} 
Check this alternative to include a legend in the chart (checked by default). It \\
is possible to change the Legend Settings (page 625) by clicking the Set- \\
tings... button. If the chart is dimensionless, but has several expressions, \\
unmarking this check box will show the expressions on the axis instead.
\end{tabular} \\
Limit Legend (Char- & \begin{tabular}{l} 
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 fol- \\
acters)
\end{tabular}\(\quad\)\begin{tabular}{l} 
lowed by ... in the chart.
\end{tabular}
\end{tabular}

In the Chart Scrolling group you can make settings for scrolling in the chart.
\begin{tabular}{ll} 
Enable X-Axis Scroll- & \begin{tabular}{l} 
Enable this check box to show a scroll control in place of the X-axis. The \\
bar \\
scroll bar can be used to scroll the selection of the X-axis values that are dis- \\
played. The number of values shown at any one time will be the number set \\
under When Number of Items Exceeds.
\end{tabular} \\
Reversed & Checking the box displays the values in reversed order.
\end{tabular}

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.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Reference Lines (page 626) dialog where you can create a new ref- \\
erence line in the chart.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to edit its \\
properties in the Reference Lines (page 626) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing reference line in the list and click this button to delete it \\
from the list.
\end{tabular}
\end{tabular}

The Text in Chart group is used for adding free-floating text to the chart.
\begin{tabular}{ll} 
Add & \begin{tabular}{l} 
Opens the Chart Text (page 627) dialog, where you can create a new chart \\
text.
\end{tabular} \\
Edit & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to edit its properties \\
in the Chart Text (page 627) dialog.
\end{tabular} \\
Delete & \begin{tabular}{l} 
Highlight an existing text in the list and click this button to delete it from the \\
list.
\end{tabular}
\end{tabular}

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 592).

\subsection*{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 628).

\subsection*{60.10 Colors}

On the Colors page you can set the display colors. This page is identical to Chart Properties: Colors (page 631).

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{61 Pivot Table}
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|l|}{Sales per CategoryName} \\
\hline [2] Category\lame & [3) ProductName & Sales & Quantity \\
\hline \multirow[t]{10}{*}{Men's Clothes} & Atles Lussekofta & \$30,126.55 & \(1057{ }^{\text {® }}\) \\
\hline & Bow tie & \$9,534.57 & 1315 \\
\hline & Desperado Jeans & \$18,240.68 & 706 \\
\hline & Lenin Jeansshorts & \$14,900.64 & 828 \\
\hline & Mr2 Trousers & \$17,944.48 & 1067 \\
\hline & O-Man Underwear & \$1,649.87 & 298 \\
\hline & Rossi Bermuda Shorts & \$10,947.25 & 1397 \\
\hline & Samba Soccer Socks & \$4,941.14 & 1175 \\
\hline & US-Master Jeans & \$21,764.94 & 817 \\
\hline & Total & \$130,050.12 & 8660 \\
\hline \multirow[t]{8}{*}{Women's Clothes} & Chantell Shirt & \$7,504.70 & 388 \\
\hline & Halter Dress & \$361,096.85 & 981 \\
\hline & Jack Flash Dress & \$42,638.00 & 722 \\
\hline & Langoste Shirt & \$4,433.35 & 246 \\
\hline & Le Baby Dress & \$47,571.88 & 623 \\
\hline & Minnki Pälsii & \$10,472.71 & 184 \\
\hline & Okkaba Skin Jackets & \$42,258.78 & 601 \\
\hline & Oyaki Kimono & \$9,084.42 & 806 \\
\hline
\end{tabular}

\section*{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 593) 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 747) from the Tools menu.
By a right-click on the pivot table the Pivot Table: Object Menu (page 718) 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 595).

\subsection*{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.

\footnotetext{
Note!
Pivoting is disabled if the Allow Pivoting check box in the Chart Properties: Presentation (Pivot Table) (page 721) 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 718) you also have access to the commands Expand all, Collapse all, Collapse Dimension Rows and Collapse Dimension Columns which effect these setting accordingly.

\section*{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 drag－ ging the mouse cursor up above the expression label row．The selected column and its target are high－ lighted in blue while you are dragging．
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{sum（Sales）} \\
\hline Company & Product & Year & Sum（Saleș \({ }^{\text {min }}\) & avg（Sales） \\
\hline \multirow[t]{7}{*}{ABC} & \multirow[t]{3}{*}{A \(\square\)} & 1998 & ，\(\sim_{3}\) & 3.0 \\
\hline & & 1999 & 2 & 2.0 \\
\hline & & Totar & 5 & 2.5 \\
\hline & \multirow{3}{*}{B} & 1998 & 1 & 1.0 \\
\hline & & 1999 & 1 & 1.0 \\
\hline & & Total & 2 & 1.0 \\
\hline & Total & & 7 & 1.8 \\
\hline \multirow[t]{7}{*}{XYZ} & \multirow[t]{3}{*}{A \(\square^{\text {a }}\)} & 1998 & 5 & 5.0 \\
\hline & & 1999 & 4 & 4.0 \\
\hline & & Total & 9 & 4.5 \\
\hline & \multirow[t]{3}{*}{日 曰} & 1998 & 7 & 7.0 \\
\hline & & 1999 & 6 & 6.0 \\
\hline & & Total & 13 & 6.5 \\
\hline & Total & & 22 & 5.5 \\
\hline Total & & & 29 & 3.6 \\
\hline
\end{tabular}

The result：
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline \multicolumn{8}{|l|}{sum（Sales）} \\
\hline & Product & A & & B & & Total & \\
\hline Company & Year & sum（Sales） & avg（Sales） & sum（Sales） & avg（Sales） & sum（Sales） & avg（Sales） \\
\hline \multirow[t]{3}{*}{ABC \({ }^{\text {a }}\)} & 1998 & 3 & 3.0 & 1 & 1.0 & 4 & 2.0 \\
\hline & 1999 & 2 & 2.0 & 1 & 1.0 & 3 & 1.5 \\
\hline & Total & 5 & 2.5 & 2 & 1.0 & 7 & 1.8 \\
\hline \multirow[t]{3}{*}{XYZ 曰} & 1998 & 5 & 5.0 & 7 & 7.0 & 12 & 6.0 \\
\hline & 1999 & 4 & 4.0 & 6 & 6.0 & 10 & 5.0 \\
\hline & Total & 9 & 4.5 & 13 & 6.5 & 22 & 5.5 \\
\hline \multicolumn{2}{|l|}{Total} & 14 & 3.5 & 15 & 3.8 & 29 & 3.6 \\
\hline
\end{tabular}

Not only the dimension fields，but also the expression row can be pivoted：
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline \multicolumn{8}{|l|}{sum（Sales）} \\
\hline & Product & A & & B & & Total & \\
\hline Company & Year & sum（Sales） & avg（Sales） & sum（Sales） & avg（Sales） & sum（Sales） & avg（Sales） \\
\hline \multirow[t]{3}{*}{ABC \({ }^{\text {a }}\)} & 1998 & 3 & 3.0 & 1 & 1.0 & 4 & 2.0 \\
\hline & 1999 & 2 & 2.0 & 1 & 1.0 & 3 & 1.5 \\
\hline & Total & 5 & 2.5 & 2 & 1.0 & 7 & 1.8 \\
\hline \multirow[t]{3}{*}{XYZ －} & 1998 & 5 & 5.0 & 7 & 7.0 & 12 & 6.0 \\
\hline & 1999 & 4 & 4.0 & 6 & 6.0 & 10 & 5.0 \\
\hline & Total & 9 & 4.5 & 13 & 6.5 & 22 & 5.5 \\
\hline \multicolumn{2}{|l|}{Total} & 14 & 3.5 & 15 & 3.8 & 29 & 3.6 \\
\hline
\end{tabular}

The result：
\begin{tabular}{|c|c|c|c|c|c|}
\hline \multicolumn{6}{|l|}{sum（Sales）} \\
\hline Company & Year & Product & A & B & Total \\
\hline \multirow[t]{6}{*}{ABC} & \multirow[b]{2}{*}{1998} & sum（Sales） & 3 & 1 & 4 \\
\hline & & avgi（Sales） & 3.0 & 1.0 & 2.0 \\
\hline & \multirow[b]{2}{*}{1999} & sum（Sales） & 2 & 1 & 3 \\
\hline & & avg（Sales） & 2.0 & 1.0 & 1.5 \\
\hline & \multirow[b]{2}{*}{Total} & sum（Sales） & 5 & 2 & 7 \\
\hline & & avg（Sales） & 2.5 & 1.0 & 1.8 \\
\hline \multirow[t]{6}{*}{¢ \({ }^{\text {® }}\)} & \multirow[b]{2}{*}{1998} & sum（Sales） & 5 & 7 & 12 \\
\hline & & avg（Sales） & 5.0 & 7.0 & 6.0 \\
\hline & \multirow[t]{2}{*}{1999} & sum（Sales） & 4 & 6 & 10 \\
\hline & & avg（Sales） & 4.0 & 6.0 & 5.0 \\
\hline & \multirow[t]{2}{*}{Total} & sum（Sales） & 9 & 13 & 22 \\
\hline & & avg（Sales） & 4.5 & 6.5 & 5.5 \\
\hline \multicolumn{2}{|l|}{\multirow[b]{2}{*}{Total}} & sum（Sales） & 14 & 15 & 29 \\
\hline & & avg（Sales） & 3.5 & 3.8 & 3.6 \\
\hline
\end{tabular}

\section*{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．

\section*{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．

\section*{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．
\begin{tabular}{|c|c|c|}
\hline \multicolumn{3}{|l|}{Sum（Sales）} \\
\hline Company & sum（Sales） & avg（Sales） \\
\hline ABC 田 & 7 & 1.8 \\
\hline XYZ 田 & 22 & 5.5 \\
\hline Total & 29 & 3.6 \\
\hline
\end{tabular}

A click on the first＋icon displays the Product details for the company ABC：
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|l|}{Suin（Sales）} \\
\hline Company & Product & sum（Sales） & avg（Sales） \\
\hline \multirow[t]{3}{*}{ABC \({ }^{\text {a }}\)} & A 田 & 5 & 2.5 \\
\hline & 日 䀦 & 2 & 1.0 \\
\hline & Total & 7 & 1.8 \\
\hline XYZ 回 & & 22 & 5.5 \\
\hline Total & & 29 & 3.6 \\
\hline
\end{tabular}

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：
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|l|}{sum（Sales）} \\
\hline Company & Product & sum（Sales） & avg（Sales） \\
\hline \multirow[t]{3}{*}{ABC \({ }^{\square}\)} & A 困 & 5 & 2.5 \\
\hline & 日 田 & 2 & 1.0 \\
\hline & Total & 7 & 1.8 \\
\hline \multirow[t]{3}{*}{XYZ －} & A 田 & 9 & 4.5 \\
\hline & B 田 & 13 & 6.5 \\
\hline & Total & 22 & 5.5 \\
\hline Total & & 29 & 3.6 \\
\hline
\end{tabular}

The + icons in the Product column indicate that another level exists．Right－click in the Product col－ umn and select Expand all again．The result will be as follows：
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{sum（Sales）} \\
\hline Company & Product & Year & sum（Sales） & avg（Sales） \\
\hline \multirow[t]{7}{*}{ABC} & \multirow[t]{3}{*}{A \({ }^{\text {a }}\)} & 1998 & 3 & 3.0 \\
\hline & & 1999 & 2 & 2.0 \\
\hline & & Total & 5 & 2.5 \\
\hline & \multirow[t]{3}{*}{日 \(\square^{\text {日 }}\)} & 1998 & 1 & 1.0 \\
\hline & & 1999 & 1 & 1.0 \\
\hline & & Total & 2 & 1.0 \\
\hline & \multicolumn{2}{|l|}{Total} & 7 & 1.8 \\
\hline \multirow[t]{7}{*}{XYZ} & \multirow[t]{3}{*}{A－} & 1998 & 5 & 5.0 \\
\hline & & 1999 & 4 & 4.0 \\
\hline & & Total & 9 & 4.5 \\
\hline & \multirow[t]{3}{*}{B \({ }^{-1}\)} & 1998 & 7 & 7.0 \\
\hline & & 1999 & 6 & 6.0 \\
\hline & & Total & 13 & 6.5 \\
\hline & \multicolumn{2}{|l|}{Total} & 22 & 5.5 \\
\hline \multicolumn{3}{|l|}{Total} & 29 & 3.6 \\
\hline
\end{tabular}

As no + icons are available in the Year column，we can conclude that there are only three dimension fields available in this pivot table．

\section*{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：
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{sum（Sales）} \\
\hline Company & Product & Year & sum（Sales） & avg（Sales） \\
\hline \multirow[t]{5}{*}{ABC} & A & 回 & 5 & 2.5 \\
\hline & \multirow{3}{*}{B} & 何 1998 & 1 & 1.0 \\
\hline & & 1999 & 1 & 1.0 \\
\hline & & Total & 2 & 1.0 \\
\hline & \multicolumn{2}{|l|}{Total} & 7 & 1.8 \\
\hline \multirow[t]{5}{*}{XYZ} & A & 回 & 9 & 4.5 \\
\hline & \multirow[t]{3}{*}{B \(\square^{\text {® }}\)} & 1998 & 7 & 7.0 \\
\hline & & 1999 & 6 & 6.0 \\
\hline & & Total & 13 & 6.5 \\
\hline & \multicolumn{2}{|l|}{Total} & 22 & 5.5 \\
\hline \multicolumn{3}{|l|}{Total} & 29 & 3.6 \\
\hline
\end{tabular}

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 Col－ lapse Dimension Columns from the Object menu．The result will be as follows：
\begin{tabular}{|r|r|r|}
\hline \multicolumn{2}{|l|}{ sum（Sales）} \\
\hline & sum（Sales） & avg（Sales） \\
\hline 目 & 29 & 3.6 \\
\hline
\end{tabular}

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．
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline \multicolumn{8}{|l|}{sum（Sales）} \\
\hline & Product & \multicolumn{2}{|l|}{A} & \multicolumn{2}{|l|}{B} & \multicolumn{2}{|l|}{Total} \\
\hline Company & Year & sum（Sales） & avg（Sales） & sum（Sales） & avg（Sales） & \multirow[t]{2}{*}{sum（Sales）} & avg（Sales） \\
\hline \multirow[t]{3}{*}{ABC \({ }^{\text {日 }}\)} & 1998 & 3 & 3.0 & 1 & 1.0 & & 2.0 \\
\hline & 1999 & 2 & 2.0 & 1 & 1.0 & 3 & 1.5 \\
\hline & Total & 5 & 2.5 & 2 & 1.0 & 7 & 1.8 \\
\hline \multirow[t]{3}{*}{XYZ \({ }^{\text {a }}\)} & 1998 & 5 & 5.0 & 7 & 7.0 & 12 & 6.0 \\
\hline & 1999 & 4 & 4.0 & 6 & 6.0 & 10 & 5.0 \\
\hline & Total & 9 & 4.5 & 13 & 6.5 & 22 & 5.5 \\
\hline Total & & 14 & 3.5 & 15 & 3.8 & 29 & 3.6 \\
\hline
\end{tabular}

\section*{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．．．

Expand all
Collapse all
Collapse Dimension
Rows
Collapse Dimension

\section*{Columns}

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．

These commands operate on the expand（＋）and collapse（－）icons that are dis－ played in multi－dimensional pivot tables．

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．
By making a copy of a pivot table and detaching it，you may make direct com－ parisons between the copy and the original．
\begin{tabular}{ll} 
Attach & \begin{tabular}{l} 
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.
\end{tabular} \\
Clone & Makes an identical copy of the pivot table. If a detached pivot table is cloned \\
the clone will be attached.
\end{tabular} Order \begin{tabular}{l} 
This cascade menu is only available when the Design Grid command of the \\
\\
View Menu (page 58) is activated or when the Always Show Design Menu \\
Items (page 86) checkbox is marked. It contains four commands to set the lay- \\
out 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
\end{tabular}
\begin{tabular}{|c|c|}
\hline Print... & Opens the Print: General (page 99) dialog from which you can print the pivot table. \\
\hline Print as PDF... & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Send to Excel & Exports the table to Microsoft Excel, which is automatically launched if not already running. The table will appear in a new Excel worksheet. For this functionality to work Microsoft Excel 97 or later must be installed on the computer. \\
\hline Export... & \begin{tabular}{l}
Opens the Save as dialog where path, file name and file type for the exported table content can specified. \\
The file formats offered include a range of delimited text file formats, HTML, XML, BIFF (native Excel format) and QVD (QlikView Data Files).
\end{tabular} \\
\hline Copy to Clipboard & \begin{tabular}{l}
This cascade menu contains the various copy options for the sheet object. \\
Full Table \\
Copies the table to the clipboard, complete with header and selection status. \\
Table Data Area \\
Copies only the values of the table to the clipboard. \\
Cell Value \\
Copies the text value of the cell right-clicked upon (when invoking the Object menu) to the clipboard. \\
Image \\
Copies an image of the sheet object to the clipboard. The image will include or exclude the sheet object caption and border depending on the settings in the User Preferences dialog, Export page. \\
Object \\
Copies the entire sheet object to the clipboard for pasting elsewhere in the layout or in another document opened within the current instance of QlikView.
\end{tabular} \\
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\square\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline
\end{tabular}

\section*{Help Opens QlikView help.}

Remove
Removes the active sheet object from the sheet.

\subsection*{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 593).

\subsection*{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 600).

\subsection*{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 609).

\subsection*{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 618).

\subsection*{61.7 Chart Properties: Presentation (Pivot Table)}


\footnotetext{
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.

\section*{Label for Totals}

Show Partial Sums
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.

Alignment
Displays partial sums in the pivot table.
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.

Vertical Text on Col- Text for the column headers will be rotated to vertical. umn Labels

Selection Indicators
With this option checked, a colored Indicator is displayed in the header of any field dimension where a selection has been made.

Always Fully This alternative means that you will not be able to collapse dimensions by Expanded clicking on the - icons.

Suppress Expansion Icons in Print

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.

Suppress Zero-Values This check box eliminates columns or rows that contain only zeros from the table.

Suppress Missing This check box eliminates columns or rows that are empty from the table.
Populate Missing Cells When this check box is marked, cells in cross tables representing missing combinations of dimensions will be mapped to a regular null value. Thereby it becomes possible to apply expressions testing for null and for attribute expressions and style formats to be applied. This setting is activated by default for all pivot tables created in QlikView 7.5 and later.

Null Symbol
Missing Symbol
The symbol entered here will be used for displaying NULL values in the table.
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.
\(\begin{array}{ll}\text { Subtotals on Top } & \text { If this option is checked, totals will be displayed top/left in the pivot table. } \\ \text { Subtotals at Bottom } & \text { If this option is checked, totals will be displayed bottom/right. }\end{array}\)

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.

Wrap Header Text If this option is checked, the contents of a label cell will be displayed in two or more rows. The Header Height _ Lines determines the number of cell lines.

Wrap Cell Text Same as above, but the setting applies to data cells. The value is set as Cell Height _ Lines.

\subsection*{61.8 Chart Properties: Visual Cues}


\section*{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 >=, Normal and Lower <= 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.

\subsection*{61.9 Chart Properties: Style}


Cell Background Color Transparency


Cell Borders Transparency


\section*{Background.}

\section*{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.
\(\left.\begin{array}{ll}\text { Current Style } & \begin{array}{l}\text { Choose an appropriate table style from the drop-down list. If the value [Cus- } \\ \text { tom] appears in the drop-down control a custom style has been applied to the } \\ \text { table. If you change the setting back to one of the pre-defined styles, the cus- } \\ \text { tom formatting will be lost. }\end{array} \\ \text { Stripes every_Rows } & \begin{array}{l}\text { Here you can specify if and at how long intervals shaded stripes should } \\ \text { appear. }\end{array} \\ \text { Indent Mode } & \begin{array}{l}\text { This setting is only valid for pivot tables. With this alternative checked, you } \\ \text { can achieve a slightly different table style that is especially useful when you } \\ \text { need to accommodate a number of dimension labels within a limited table } \\ \text { width. } \\ \text { Use Only First Dimension Label } \\ \text { This setting is only available for pivot tables already in Indent Mode and mod- } \\ \text { ifies the style of the pivot table further. }\end{array} \\ \text { Vertical Dimension } & \begin{array}{l}\text { This setting determines whether vertical cell borders are displayed for dimen- } \\ \text { sion columns. }\end{array} \\ \text { Cell Borders } & \begin{array}{l}\text { As above, but for expression columns. }\end{array} \\ \text { Vertical Expression } & \begin{array}{l}\text { Provided that a Spacing has been determined in the Advanced Field Settings }\end{array} \\ \text { Border Above Spacing } \\ \text { Bage 496) dialog, the table style can be slightly modified by checking this }\end{array}\right\}\)

\section*{Cell Borders Trans- Sets how pronounced the cell borders should be.} parency

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{62 Straight Table}
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|l|}{Sales per CategoryName} \\
\hline [2] Category*ame & (2) ProductName & Sales & Quantity \\
\hline & & \$1,565,525.31 & 51952 \\
\hline Men's Clothes & Atles Lussekofta & \$30,126.55 & 1057 \\
\hline Men's Clothes & Bow tie & \$9,534.57 & 1315 \\
\hline Men's Clothes & Desperado Jeans & \$18,240.68 & 706 \\
\hline Men's Clothes & Lenin Jeansshorts & \$14,900.64 & 828 \\
\hline Men's Clothes & Mr2 Trousers & \$17,944.48 & 1067 \\
\hline Men's Clothes & O-Man Underwear & \$1,649.87 & 298 \\
\hline Men's Clothes & Rossi Bermuda Shorts & \$10,947.25 & 1397 \\
\hline Men's Clothes & Samba Soccer Socks & \$4,941.14 & 1175 \\
\hline Men's Clothes & US-Master Jeans & \$21,764.94 & 817 \\
\hline Women's Clothes & Chantell Shirt & \$7,504.70 & 388 \\
\hline Women's Clothes & Halter Dress & \$361,096.85 & 981 \\
\hline Women's Clothes & Jack Flash Dress & \$42,638.00 & 722 \\
\hline Women's Clothes & Langoste Shirt & \$4,433.35 & 246 \\
\hline Women's Clothes & Le Baby Dress & \$47,571.88 & 623 \\
\hline Women's Clothes & Minnki Pälsii & \$10,472.71 & 184 \\
\hline Women's Clothes & Okkaba Skin Jackets & \$42,258.78 & 601 \\
\hline Women's Clothes & Oyaki Kimono & \$9,084.42 & 806 \\
\hline
\end{tabular}

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 593) 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 747) from the Tools menu.
By a right-click on the straight table the Straight Table: Object Menu (page 727) 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 595).

\subsection*{62.1 Using the Straight Table}

\section*{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.

\section*{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.

\subsection*{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:
\begin{tabular}{ll} 
Properties... & \begin{tabular}{l} 
Opens the Properties dialog where the parameters defining the straight table \\
can be set.
\end{tabular} \\
Detach & \begin{tabular}{l} 
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. \\
By making a copy of a straight table and detaching it, you may make direct \\
comparisons between the copy and the original.
\end{tabular} \\
Attach \begin{tabular}{l} 
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.
\end{tabular} \\
Clone Columns to Data \(\quad\)\begin{tabular}{l} 
Makes an identical copy of the straight table. If a detached straight table is \\
cloned the clone will be attached.
\end{tabular} \\
Adjusts the width of each table column to its longest data string. Headers are \\
included in the calculations.
\end{tabular}

\section*{any sheet object on the current sheet.}

\section*{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.

\section*{Bring Forward}

Increases the layout layer of the sheet object by one. Maximum value is 127 .

\section*{Send Backward}

Decreases the layout layer of the sheet object by one. Minimum value is -128 .
\begin{tabular}{ll} 
Change Value & \begin{tabular}{l} 
Only available for expression columns containing an inputsum aggregation of \\
an input field. Sets the cell clicked on in input edit mode. Equivalent to click- \\
ing the input icon in the cell.
\end{tabular} \\
Restore Values & \begin{tabular}{l} 
Only available for expression columns containing an inputsum aggregation of \\
an input field. Opens a cascade menu with three options. \\
Restore Single Value \\
Restores the field values underlying the cell clicked upon on to their default \\
values from the script. \\
Restore Possible Values \\
Restores the values of all possible underlying field values to their default \\
values from the script. \\
Restore All Values
\end{tabular} \\
Restores the values of all underlying field values to their default values from \\
the script.
\end{tabular}\(\quad\)\begin{tabular}{l} 
Clears all selections in the dimensions and expressions of the table.
\end{tabular}
\begin{tabular}{|c|c|}
\hline Linked Objects & \begin{tabular}{l}
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.
\end{tabular} \\
\hline Minimize & Iconizes the object. Clicking \(\square\) 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 479) page. \\
\hline Maximize & Enlarges the object to fill the screen. Clicking 回 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 479) page. \\
\hline Restore & Restores a minimized or maximized object to its previous size and location. Double-clicking the icon of a minimized object or clicking 圆 in the object caption (if shown) of a maximized object produces the same result. This command is available only for minimized or maximized objects. \\
\hline Help & Opens QlikView help. \\
\hline Remove & Removes the active sheet object from the sheet. \\
\hline \multicolumn{2}{|l|}{62.3 General} \\
\hline \multicolumn{2}{|l|}{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 593).} \\
\hline
\end{tabular}

\subsection*{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 600).

\subsection*{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 605).

\subsection*{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 609).

\subsection*{62.7 Chart Properties: Sort (Straight Table)}


\section*{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.

\section*{Expression}

Sorts the column values according to the expression entered into the text edit box below this sort option.

\section*{Frequency}

Sorts the column values by frequency (number of occurrences in the table).

\section*{Numeric value}

Sorts the column values by their numeric values.

\section*{Text}

Sorts the column values in alphabetical order.

\section*{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.

\subsection*{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 731)page.
\begin{tabular}{ll} 
Conditional & \begin{tabular}{l} 
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 \\
Edit Expression Dialog (page 737) is opened for easier editing of long for- \\
mulas
\end{tabular} \\
Alignment & \begin{tabular}{l} 
In this group, the alignment of the dimension values, expression values and \\
their labels within the straight 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.
\end{tabular} \\
Dropdown Select & \begin{tabular}{l} 
The Dropdown Select 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.
\end{tabular} \\
Searchable \\
If the checkbox Searchable is selected for an expression column, a search icon
\end{tabular}
\begin{tabular}{ll} 
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.
\end{tabular}

\subsection*{62.9 Chart Properties: Visual Cues}


\section*{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 >=, Normal and Lower <= 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.

\subsection*{62.10 Chart Properties: Style}


Cell Background Color Transparency
\(0 \% \square 100 \%\)
Cell Borders Transparency
\(0 \% \square 100 \%\)
\(\square\) Border Above Spacing

\section*{Background..}

\section*{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.
\begin{tabular}{ll} 
Current Style & \begin{tabular}{l} 
Choose an appropriate table style from the drop-down list. If the value [Cus- \\
tom] 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 cus- \\
tom formatting will be lost.
\end{tabular} \\
Stripes every_Rows & \begin{tabular}{l} 
Here you can specify if and at how long intervals shaded stripes should \\
appear.
\end{tabular} \\
Indent Mode & \begin{tabular}{l} 
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. \\
Use Only First Dimension Label
\end{tabular} \\
This setting is only available for pivot tables already in Indent Mode and mod- \\
ifies the style of the pivot table further.
\end{tabular}

Background... Opens the Background Settings (page 472) dialog.
Cell Background If a color or an image has been applied in Background Settings, you can Color Transparency adjust the transparency of that color or image in the cell background here.

Cell Borders Trans- Sets how pronounced the cell borders should be. parency

\subsection*{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 634).

\subsection*{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 475).

\subsection*{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 476).

\subsection*{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 479).

\section*{63 Edit Expression Dialog}


\section*{The Edit Expression dialog}

This dialog is opened by selecting the Add button below the expressions list in Chart Properties: Expressions (page 609) or by right-clicking an existing expression and selecting Add Expression or Edit Expression.
Though it is primarily use is chart expressions, the same dialog is also utilized for building or editing other expressions wherever Calculated Formula (page 851) 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.

\section*{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.

\section*{Fields}


The Fields tab holds controls that are used for creating statistical aggregation functions based on field data.
\begin{tabular}{ll} 
Aggregation & \begin{tabular}{l} 
The first drop down contains a list of available statistical aggregation func- \\
tions. (The functions First String and Last String can e.g. be used for showing \\
text in a pivot table.)
\end{tabular} \\
Table & \begin{tabular}{l} 
This drop-down list allows you to limit the fields displayed in the Field list \\
below to one specific internal table.
\end{tabular} \\
Field & \begin{tabular}{l} 
This drop down contains a list of available field names.
\end{tabular} \\
Show System Fields & \begin{tabular}{l} 
Check this alternative in order to include the system fields in the Field list.
\end{tabular} \\
Distinct & \begin{tabular}{l} 
Statistical functions are by default calculated on number of occurrences. By \\
checking the Distinct option, the expression will be calculated without dupli- \\
cate values. (For a better explanation of the use of the distinct qualifier, see \\
the Chart Aggregation functions (page 770)).
\end{tabular} \\
Paste & \begin{tabular}{l} 
Click this button in order to enter your selections as a component of the \\
expression.
\end{tabular}
\end{tabular}

\section*{Functions}


The Functions tab holds controls that are used for entering general QlikView-functions into the expression.
\begin{tabular}{ll} 
Function Category & \begin{tabular}{l} 
This drop-down list allows you to limit the functions displayed in the Func- \\
tion Name list below to a single category at a time.
\end{tabular} \\
Function Name & \begin{tabular}{l} 
This drop down contains a list of all available QlikView-functions (except a \\
limited number of dedicated script functions).
\end{tabular} \\
Paste & \begin{tabular}{l} 
The pane at the bottom of the functions tab displays the argument syntax of \\
the selected function.
\end{tabular} \\
Click this button in order to enter your selections as a component of the \\
expression.
\end{tabular}

\section*{Variables}


The Variables tab holds controls that are used for entering QlikView-variables into the expression.

\footnotetext{
Variables This drop down contains a list of all variables currently defined for the doc-
} ument.

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.

\section*{Images}


The Images tab holds controls that are used for accessing the Internal Files (page 165) 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.
\begin{tabular}{ll} 
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... & \begin{tabular}{l} 
Click this button to select directly from the pictures in the Image Selector \\
dialog.
\end{tabular} \\
Paste & \begin{tabular}{l} 
Click this button in order to enter your selections as a component of the \\
expression.
\end{tabular}
\end{tabular}

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.

\subsection*{63.1 File Menu}

\section*{Export to Expression}

\section*{File...}

\section*{Insert File...}

\section*{Colormix Wizard...}

Print

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 .que.

A previously created expression file can be inserted into the script by means of the Insert File... command.

A color mix expression can be created using the Colormix Wizard (page 740).
Opens Windows standard Print dialog for printing the expressions. This command can also be invoked by the following keyboard shortcut: CTRL +P .

\subsection*{63.2 Edit Menu}
\begin{tabular}{ll} 
Undo & \begin{tabular}{l} 
Undoes the latest change. This command can also be invoked by the fol- \\
lowing keyboard shortcut: CTRL+Z.
\end{tabular} \\
Redo & \begin{tabular}{l} 
Redoes the latest Undo. This command can also be invoked by the following \\
keyboard shortcut: CTRL+Y.
\end{tabular} \\
Cut & \begin{tabular}{l} 
Exports the highlighted text to the clipboard. This command can also be \\
invoked by the following keyboard shortcut: CTRL+X.
\end{tabular} \\
Copy & \begin{tabular}{l} 
Copies the highlighted text to the clipboard. This command can also be \\
invoked by the following keyboard shortcut: CTRL+C.
\end{tabular} \\
Paste & \begin{tabular}{l} 
Pastes the contents of the clipboard into the dialog at the position of the cur- \\
sor. This command can also be invoked by the following keyboard shortcut: \\
CTRL+V.
\end{tabular} \\
Clear & \begin{tabular}{l} 
Clears the entire expression.
\end{tabular} \\
Find/Replace... & \begin{tabular}{l} 
Selects the entire expression.
\end{tabular} \\
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.
\end{tabular}

\subsection*{63.3 Settings Menu}

\author{
Configure... Opens User Preferences: Editor (page 82).
}

\subsection*{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.

\section*{Step 1 - Enter a Value Expression}


This is the expression that will control what color QlikView will show. Typical expressions could be
- Sum(Sales) / Sum(total Sales)
- Sum(Sales) / Sum(Quota)
- Avg(Age)

Note that it is usually a value that has the same order of magnitude no matter how many or how few records that you have selected. An average, a percentage or an index is usually a good measure to use.
Click Next to continue.

\section*{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:
\begin{tabular}{lll} 
Value Expression & Upper Limit & Lower Limit \\
\begin{tabular}{l} 
Sum(Sales) / Sum(total \\
Sales)
\end{tabular} & Sum(total Sales) & 0 \\
Sum(Sales) / Sum(Quota) & \(2(=200 \%)\) & 0 \\
Avg(Age) & Max(total Age) & Min(total Age)
\end{tabular}

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.
\begin{tabular}{ll} 
Auto Normalize & \begin{tabular}{l} 
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.
\end{tabular} \\
Upper Limit & \begin{tabular}{l} 
Here an expression for an upper limit must be entered, unless the Auto Nor- \\
malize option is checked.
\end{tabular} \\
Intermediate & \begin{tabular}{l} 
Here it is possible to enter an expression for an intermediate level, linked to a \\
third color.
\end{tabular} \\
Lower Limit & \begin{tabular}{l} 
Here an expression for a lower limit must be entered, unless the Auto Nor- \\
malize option is checked.
\end{tabular} \\
Reverse & This button reverses the colors for the upper and lower limits.
\end{tabular}

Click Next to continue.

\section*{Step 3 - Finalize}
\begin{tabular}{|c|c|}
\hline Step 3-Colormix & \(\pm\) \\
\hline \begin{tabular}{l}
V Enhanced Colors \\
Value Saturation
Use Upper (Lower) Color
Use Standard Color
\end{tabular} & \begin{tabular}{l}
Enhanced colors will increase the sensitivity in the midrange so that the color is pushed towards the Upper and Lower colors. \\
Value Saturation defines what color to show if the value exceeds the upper (or lower) limit as defined in Step 2 of this wizard. \\
If Use Standard Color is marked, Qlik View will revert to the standard colors as defined on the Colors page.
\end{tabular} \\
\hline
\end{tabular}

In this page you finalize the color mix expression.

\section*{Enhanced 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.

\section*{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.

\subsection*{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.

\subsection*{64.2 Boxplot Wizard Presentation}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline \multicolumn{6}{|l|}{Step 2 - Presentation} & \(\underline{x}\) \\
\hline Display Mode & \multicolumn{6}{|c|}{Display Mode} \\
\hline \multicolumn{7}{|l|}{Average mode} \\
\hline & Includ Mark inclu & iskers heck box in the b & se how & he whisk & & \\
\hline V Include Whiskers & \multicolumn{6}{|c|}{\begin{tabular}{l}
Use Outliers \\
Mark this check box to show outliers for extreme values.
\end{tabular}} \\
\hline Min/Max & - Click & to compl & zard. & & & \\
\hline \multicolumn{7}{|l|}{\(\square\) Use Outliers (Min/Max)} \\
\hline
\end{tabular}

\section*{Display Mode}

\section*{Include Whiskers}

Use Outliers (Min/Max)

Choose the Display Mode for the distribution, Average or Median.
Include whiskers to have the chart display the upper and lower whiskers, as either Min/Max or \(\mathbf{5} / \mathbf{9 5}\) percentiles.

Mark this check box to show so called outliers for extreme values.

\section*{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.

\subsection*{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 dimensons.
Select one or more dimensions in the drop-down boxes. The dimension(s) define the values for which to calculate the chart expres-
sion(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.

\section*{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.

\section*{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.

\section*{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.

\section*{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.


\section*{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.

\section*{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.

\section*{Mode}

Show data points as lines, symbols or a combination thereof. Make the selection by clicking on the appropriate icon.

\section*{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.
\begin{tabular}{lll} 
Step 4 - Chart format & \begin{tabular}{l} 
Style \\
Select a pie chart style by clicking on one of the icons. \\
Max Number of Slices (2-50)
\end{tabular} \\
Selecct the maximum number individual slices in the pie. Remaining data points \\
will be collected in an Others' segment. \\
Show Numbers in Legend \\
Mark this check boxify fou want the expression values to be shown as \\
numbers in the chart legend.
\end{tabular}

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.
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline \multicolumn{6}{|l|}{Step 4 - Chart format} & \(x^{x}\) \\
\hline Style & \multicolumn{6}{|c|}{\multirow[t]{2}{*}{\begin{tabular}{l}
Style \\
Select a table style from the drop-down box.
\end{tabular}}} \\
\hline Table 1 & & & & & & \\
\hline
\end{tabular}

\section*{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.


\section*{Step 4: Chart format (pivot table)}

In this page you set formatting options for the selected chart type.

\section*{Style}

Select a table style from the drop-down box.

\section*{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.

\section*{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.

\section*{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.

\section*{66 Time Chart Wizard}

The time chart wizard helps you to 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.

\subsection*{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:


\section*{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.


\section*{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.


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).

\section*{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.

\section*{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.
\begin{tabular}{|c|c|c|}
\hline Step 4 - Select time & eriod & \(x^{x}\) \\
\hline Time Period
Year
Quarter
Month(s)
Week
Lunar week
Day & \begin{tabular}{l}
Multiple Months:
\(\square\) \\
1
\end{tabular} & \begin{tabular}{l}
Time Period \\
Select the length of the period by which you want to do comparisons. Time periods of 1, 2, 3, 4 or 6 months can be selected in the Multiple Months drop-down. \\
To Date Mode \\
Mark this check box 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.
\end{tabular} \\
\hline \multicolumn{3}{|l|}{\(\square\) Io Date Mode} \\
\hline & < Back & Next > Finish Cancel \(\quad\) Help \\
\hline
\end{tabular}

\section*{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.


\section*{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:

\section*{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.

\section*{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.
\begin{tabular}{|c|c|}
\hline Step 6 - Define comparison period & - \\
\hline Previous Period
Year
Dynamic Comparison Year
Period
Last Completed Period as Base & \begin{tabular}{l}
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 a comparison between current year (e.g. 2006) and a non-consequtive year (e.g. 2003). you may use this option and select 2006 and 2003 in the application. This option is only available when the basic comparison period is year. \\
Last Completed Period as Base \\
With this check box marked will use the last completed period as base. e.g. the last complete quarter.
\end{tabular} \\
\hline < Back & Next > Finish Cancel Help \\
\hline
\end{tabular}

\section*{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.

\section*{Year}

Make the comparison period the same period in the previous year, e.g. same quarter last year.

\section*{Period}

Make the comparison period the previous period, e.g. 2nd quarter before 3rd quarter in a given year.

\section*{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.


\section*{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:

\section*{Include Comparison Period}

Mark this check-box in order to add a column showing the chart's base expression within the comparison period.

\section*{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.

\section*{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.

\section*{67 Statistics Chart Wizard}

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

\subsection*{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 785) and Statistical Test Functions in Script (page 295). For further information about statistical tests and their uses, please refer to a statistics text book.

\subsection*{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:
\begin{tabular}{ll} 
Column & Specify a field or expression returning column headers for the test data sample. \\
Row & \begin{tabular}{l} 
Specify a field or expression returning row headers for the test data sample.
\end{tabular} \\
Value & \begin{tabular}{l} 
Specify a field or expression returning test value data.
\end{tabular} \\
Expected Value & \begin{tabular}{l} 
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.
\end{tabular} \\
\begin{tabular}{l} 
p (Show p-value / Sig- \\
nificance)
\end{tabular} & \begin{tabular}{l} 
Mark this check box to include the palue (significance) in the resulting \\
chart.
\end{tabular}
\end{tabular}

\section*{df (Show Degrees of} Freedom)

Chi2 (Show Chi2 Value)

Mark this check box to include the df value (degrees of freedom) in the resulting chart.

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.

\subsection*{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 / Expres- \(\quad\) Specify a field or expression returning test value data series.
sion
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)
p (Show p-value / Sig-
nificance (2-tailed))
Mean difference

Lower (CL \%)

Upper (CL \%)

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.

\subsection*{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:
Grouping Field/Ex- Specify a field or expression returning test grouping data series. Exactly two pression

\section*{Test Field/Expression}

Specify a field or expression returning the test value data series.
t (Show t Value)
Mark this check box to include the t value in the resulting chart.
df (Show Degrees of Freedom)
p (Show p-value / Sig-
nificance (2-tailed))
Mean difference values must be returned.

Mark this check box to include the df value (degrees of freedom) in the resulting chart.

Mark this check box to include the p-value (significance, 2-tailed) in the resulting chart.

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.

\section*{68 Chart Expressions}

Expressions are used in charts to define calculated data, e.g. the data shown on the \(y\)-axes 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 737), 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.

\section*{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:
\begin{tabular}{ll} 
expression \(::=\) ( constant & \(\mid\) \\
expressionname & \(\mid\) \\
operatorl expression & \(\mid\) \\
expression operator2 expression & \(\mid\) \\
function & | \\
aggregation function & \\
(expression )
\end{tabular}
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.
operatorl is a unary operator (working on one expression, the one to the right).
operator 2 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 )
parameters 2 ::= aggrexpression {, aggrexpression }

```

The number and types of parameters is not arbitrary. It depends on the function used.

General syntax for aggregation:
```

aggrexpression ::= (fieldref |
operatorl aggrexpression |
aggrexpression operator2 aggrexpression |
functioninaggr |
(aggrexpression ) )

```
fieldref is a field name.

\section*{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.

\subsection*{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 799).
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.

\section*{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.

\section*{Example:}
\(\boldsymbol{\operatorname { s u m }}(\{[\) 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.

\subsection*{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:

\section*{\$( variablename )}
\$(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:

\section*{\$( \#variablename )}
\$( \#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.

\section*{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.

\subsection*{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:
\begin{tabular}{ll} 
Numeric Operators & page 768 \\
String Operators & page 768 \\
Logical Operators & page 768 \\
Relational Operators & page 768 \\
Bit Operators & page 769
\end{tabular}

\section*{Numeric Operators}

All numeric operators use the numeric values of the operands and return a numeric value as result.
\[
\begin{array}{ll}
+ & \begin{array}{l}
\text { Sign for positive number (unary operator) or arithmetic addition. The binary } \\
\text { operation returns the sum of the two operands. }
\end{array} \\
- & \begin{array}{l}
\text { Sign for negative number (unary operator) or arithmetic subtraction. The unary } \\
\text { operation returns the operand multiplied by }-1, \text { and the binary the difference } \\
\text { between the two operands. }
\end{array} \\
* & \begin{array}{l}
\text { Arithmetic multiplication. The operation returns the product of the two oper- } \\
\text { ands. }
\end{array} \\
\text { Arithmetic division. The operation returns the ratio between the two operands. }
\end{array}
\]

\section*{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.
```

\&
String concatenation. The operation returns a text string, that consists of the
two operand strings, one after another.

```

\section*{Example:}
```

'abc' \& 'xyz' returns 'abcxyz'
like String comparison with wildcard characters. The operation returns a boolean true (-1) if the string before the operator is matched by the string after the operator. The second string may contain the wildcard characters * (any number of arbitrary characters) or ? (one arbitrary character).
Examples:
'abc' like 'a*' returns true (-1)
'abcd' like 'a?c*' returns true ( -1 )
'abc' like 'a??bc' returns false (0)

```

\section*{Logical Operators}

All logical operators interpret the operands logically and return true ( -1 ) or false (0) as result.
```

not Logical inverse. One of the few unary operators. The operation returns the log- ical inverse of the operand.
Logical and. The operation returns the logical and of the operands.
or
Xor
Logical or. The operation returns the logical or of the operands.
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.

```

\section*{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.
\begin{tabular}{|c|c|c|}
\hline < & 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. \\
\hline <= & 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. \\
\hline > & 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. \\
\hline >= & 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. \\
\hline = & 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. \\
\hline <> & 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. \\
\hline precedes & ASCII less than & \begin{tabular}{l}
Unlike the < 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. \\
Example: \\
' 11' precedes ' 2'returns true compare this to: \\
' 11 ' < ' \(2^{\prime}\) returns false
\end{tabular} \\
\hline
\end{tabular}
follows ASCII greater than Unlike the \(>\) 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.

\section*{Example:}
' 23' follows ' 111' returns true compare this to:
' \(23^{\prime}>\) ' \(111^{\prime}\) returns false

\section*{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.
\begin{tabular}{lll} 
bitand & Bit and. & The operation returns the logical AND of the operands performed bit by bit. \\
bitor & Bit or. & The operation returns the logical OR of the operands performed bit by bit. \\
bitxor & \begin{tabular}{l} 
Bit exclusive \\
or.
\end{tabular} & \begin{tabular}{l} 
The operation returns the logical exclusive or of the operands performed bit \\
by bit.
\end{tabular} \\
\(\gg\) & Bit right shift. & \begin{tabular}{l} 
Unary operator. The operation returns the operand shifted one step to the \\
right.
\end{tabular} \\
\(\ll\) & Bit left shift. & Unary operator. The operation returns the operand shifted one step to the left.
\end{tabular}

\subsection*{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:
\begin{tabular}{ll} 
Basic Aggregation Functions & page 771 \\
String Aggregation Functions & page 772 \\
Counter Aggregation Functions & page 773 \\
Statistical Aggregation Functions in Charts & page 774 \\
Financial Aggregation Functions in Charts & page 783 \\
Statistical Distribution Functions & page 785 \\
Special Input Field Aggregation Functions & page 797 \\
Advanced Aggregation & page 289 \\
Set Analysis & page 799 \\
Alternate States & page 122
\end{tabular}

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 \(\mathbf{~} \mathbf{1}\}\) 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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602). See also the examples of Nested Aggregations and Related Issues (page 845).
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 799).
For a better understanding of how to use the aggregate qualifier ( total), see Examples (page 831).

\section*{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).

\section*{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.

\section*{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.

\section*{Examples:}
```

max( Sales )
max( Sales, 2 )
max( Price*Quantity )
max( total Sales )
max( {1} total Sales )

```
only([\{set_expression\}][ distinct ] [ total [<fld \{, fld\}>]] expression)

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.

\section*{Examples:}
```

only( Sales )
only( Price*Quantity )
only( total Salesman )

```

\section*{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.

\section*{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 nth value in order.

\section*{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 )

```

\section*{String Aggregation Functions}

\section*{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.

\section*{Examples:}
```

MinString( Currency )
MinString( Left( abc,2 ) )
MinString( total Currency)
MinString( <X> Currency )

```

\section*{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.

\section*{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.

\section*{Examples:}
```

concat( Code, ';' )
concat( FirstName\&' '\&LastName, ',' )
concat( distinct Code, ';' )
concat( total Name, ';' , Date )
concat( total <Grp> Name, ';' , Date)

```

\section*{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.

\section*{Examples:}
```

coung(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).

\section*{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).

\section*{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).

\section*{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.

\section*{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

```

\section*{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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).
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.

\section*{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 ( \(\mathbf{s t d e v} / \mathbf{s q r t}(n)\) ) 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 \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The sterr function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression \(\}]\) [distinct ] [ total [<fld \(\{\), fld \(\}>]]\) expression) (page 774)function.

\section*{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 \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The steyx function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774)function.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression \(\}\) ] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_m function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{Examples:}
```

linest_m( Y, X )
linest_m(A/B, X*Y/3)
linest_m( total $\mathrm{Y}, \mathrm{X})$
linest_m( total $<\mathrm{Z}>\mathrm{Y}, \mathrm{X}$ )
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 \(y=m x+b\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression \}] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_b function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 r 2 value (coefficient of determination) of a linear regression defined by the equation \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y 0 may be stated forcing the regression line to pass through the y -axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression \(\}\) ] [ distinct ] [ total \([<f l d\{, f l d\}>]]\) expression) (page 774) function. The linest_r2 function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression \(\}]\) [distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression \(\}\) ] [ distinct ] [ total \([<\) fld \(\{\), fld \(\}>]]\) expression) (page 774) function. The linest_sem function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \(\}>\) ] \(]\) expression) (page 774)function.

\section*{Examples:}
```

linest_sem( Y, X )
linest_sem ( $\mathrm{A} / \mathrm{B}, \mathrm{X} * \mathrm{Y} / 3$ )
linest_sem( total Y, X )
linest_sem( total $<\mathrm{Z}>\mathrm{Y}, \mathrm{X}$ )
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 \(y=m x+b\) 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 valye y0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression \(\}\) ] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_seb function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression \(\}]\) [distinct ] [ total [<fld \{, fld \} >]l expression) (page 774)function.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye \(y 0\) may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression\}] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_sey function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression \(\}\) ] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_df function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression \(\}]\) [distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 (r2/(1-r2)) of a linear regression defined by the equation \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y 0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression\}] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_f function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate.
Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression\}] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_ssreg function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \(\}>\) ]] expression) (page 774)function.

\section*{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 \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) 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 valye y0 may be stated forcing the regression line to pass through the \(y\)-axis at a given point. By stating both y 0 and x 0 it is possible to force the regression line to pass through a single fixed coordinate. Unless both y 0 and x 0 are stated, the function requires at least two valid data-pairs to calculate. If y 0 and x 0 are stated, a single data pair will do.
This function has the same limitations for nested aggregation as the avg([\{set_expression\}] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The linest_ssresid function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \(\}>\) l] expression) (page 774) function.

\section*{Examples:}
linest_ssresid( \(\mathrm{Y}, \mathrm{X}\) )
linest_ssresid( \(\mathrm{A} / \mathrm{B}, \mathrm{X} * \mathrm{Y} / 3\) )
linest_ssresid( total Y, X )
linest_ssresid( total \(<\mathrm{Z}>\mathrm{Y}, \mathrm{X}\) )
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 rangecorrel( \(x\)-value , \(y\)-value \(\{, x\)-value , \(y\)-value \(\}\) ) (page 815 ) 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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).

This function has the same limitations for nested aggregation as the avg([\{set_expression\}] [ distinct ] [ total \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The correl function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The median function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression \(\}]\) [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The stdev function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The skew function supports Set Analysis (page 799) and the total qualifier in the same way as the \(\operatorname{avg}([\{\) set_expression \(\}]\) [ distinct ] [ total [<fld \(\{\), fld \(\}>]]\) expression) (page 774) function.

\section*{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 \([<f l d\{\), fld \(\}>]]\) expression) (page 774) function. The kurtosis function supports Set Analysis (page 799) and the total qualifier in the same way as the avg([\{set_expression\}] [ distinct ] [ total [<fld \{, fld \} >]] expression) (page 774) function.

\section*{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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).

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.

\section*{Examples:}
```

fractile( Sales, 0.75 )
fractile( X*Y/3, 0.9 )
fractile( total Price, 0.25 )
fractile( total <Group> Price )

```

\section*{Financial Aggregation Functions in Charts}

\section*{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 311) 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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).
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.

\section*{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 312) 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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).

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.

\section*{Examples:}
```

xirr( Payments, Dates )
xirr( A/B, X*Y/3 )
xirr( total Payments, Dates )
xirr( total <Grp> Payments, Dates)

```

\section*{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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602). 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.

\section*{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, valu-
eexpression, 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 312) 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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).
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.

\section*{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)

```

\section*{Statistical Test Functions in Charts}

The following three functions apply to chi2-tests.
```

chi2test_p ([set_expression][ total [<Eld {,fld}>] col, row,
observed_value [, expected_value])

```
returns the aggregated chi \({ }^{2}\)-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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).

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.

\section*{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 \(\}\rangle\) ] col, row, observed_value [, expected_value]) (page 785) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{\) fld \(\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 785) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \} \(\gg\) ] group, value [, eq_ var \(=\) true]) (page 786) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([\langle f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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_ \(v a r=t r u e])\) (page 786) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<\) fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 786) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 786) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 786) 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\} \(\gg]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 786) and TTest_conf ([set_expression][ total [<fld \{,fld \(\}>]\) group, value [, sig \(=\) 0.025 [, eq_var \(=\) true]]) (page 788) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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\} \(\gg]\) group, value [, eq_var \(=\) true]) (page 786) and TTest_conf ([set_expression][ total [<fld \(\{, f l d\}>]\) group, value [, sig \(=\) 0.025 [, eq_var = true]]) (page 788) functions for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<\) fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 788) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle\) ] col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \(\{, f l d\}\rangle]\) weight, group, value [, eq_var = true]) (page 788) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \{,fld\} \(\rangle>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \(\}\rangle\) ] weight, group, value [, eq_var = true]) (page 788) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{\) fld \(\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \(\{, f l d\}\rangle]\) weight, group, value [, eq_var = true]) (page 788) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [<fld \{ffld\}>] col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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\} \(\rangle>\) ] weight, group, value [, eq_var \(=\) true]) (page 788) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 788) and TTestw_conf ([set_expression][ total [<fld \{,fld \(\}>]\) weight, group, value [, sig \(=0.025\) [, eq_var \(=\) true]]) (page 790) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 788) and TTestw_conf ([set_expression][ total [<fld \{,fld \(\}>]\) weight, group, value [, sig \(=0.025\) [, eq_var \(=\) true]]) (page 790) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([\langle f l d\{\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
```

TTestw_upper( Weight, Group, Value )
TTestw_upper( Weight, Group, Value, false )

```

The following eight functions apply to one-sample student's t-tests.

\section*{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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Example:}
ttest1_t( Value )

\section*{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 791) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \{,fld\} \(\gg]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 TTestl_t ([set_expression][ total [<fld \{,fld \(\}>]\) value) (page 791) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 TTestl_t ([set_expression][ total [<fld \{,fld \} \(>]\) value) (page 791) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([\langle f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Example:}
```

ttest1_dif( Value )

```

\section*{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 TTestl_t ([set_expression][ total [<fld \{,fld \(\}>]\) value) (page 791) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Example:}
ttesti_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 791) 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\} \(\rangle>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \} \(\gg]\) value) (page 791) and TTest1_conf ([set_expression][ total [<fld \{,fld \(\}>]\) value [, sig \(=0.025\) ]) (page 792) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 791) and TTestl_conf ([set_expression][ total [<fld \{,fld \(\}>]\) value [, sig \(=0.025\) ]) (page 792) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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.

\section*{TTestlw_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 \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 TTestlw_t ([set_expression][ total \([<f l d ~\{, f l d\}\rangle]\) weight, value) (page 793) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
```

    ttest1w_df( Weight, Value )
    TTestlw_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 TTestlw_t ([set_expression][ total [<fld \{,fld \(\}>]\) weight, value) (page 793) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
ttest1w_sig( Weight, Value )
TTestlw_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 TTestlw_t ([set_expression][ total [<fld \{,fld \(\}>]\) weight, value) (page 793) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [ \(\langle\) fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
ttest1w_dif( Weight, Value )
```

TTestlw_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 TTestlw_t ([set_expression][ total [<fld \{,fld\} \(\gg\) weight, value) (page 793) function for argument descriptions.
The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [<fld \{,fld\} \(\rangle\) ] col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\) \{,fld \(\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 TTestlw_t ([set_expression][ total [<fld \{,fld \(\}>]\) weight, value) (page 793) 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 \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
```

ttest1w_conf( Weight, Value )
ttest1w_conf( Weight, Value, 0.005 )

```
```

TTestlw_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 TTestlw_t ([set_expression][ total [<fld \{,fld \}>] weight, value) (page 793) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<\) fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
```

ttest1w_lower( Weight, Value )
ttest1w_lower( Weight, Value, 0.005 )

```
```

TTest1w_upper ([set_expression][ total [<fld {,fld}>] weight, value
[, sig = 0.025 ])

```
returns the aggregated value for the upper end of the confidence interval for a series of values iterated over the chart dimension(s). See the TTestlw_t ([set_expression][ total [<fld \{,fld \(\}>]\) weight, value) (page 793) 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 \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test \(p\) ([set_expression][ total \([<f l d\{\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
```

ttestlw_upper( Weight, Value )
ttest1w_upper( Weight, Value, 0.005 )

```

The following five functions apply to \(z\)-tests.
```

ZTest_z ([set_expression][ total [<fld {,fld}>] value [, sigma])

```
returns the aggregated \(z\) value for a series of values iterated over the chart dimension(s). The values should be returned by value. A population mean of 0 is assumed. If you want the test to be performed around another mean, subtract that value from the sample values. If known, the standard deviation can be stated in sigma. If sigma is omitted the actual sample standard deviation will be used. Text values, null values and missing values in value will result in the function returning null.
The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [<fld \{,fld\} \(\rangle>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 795) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 795) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 795) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\) \{,fld \(\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 795) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 \([<f l d\{\) \{fld \(\}\rangle]\) weight, value [, sigma]) (page 796) function for argument descriptions.

The same limitations with regard to nested aggregation as those listed under the chi2test \(p\) ([set_expression][ total [<fld \(\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([\langle f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 796) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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\} \(\gg]\) weight, value [, sigma]) (page 796) 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 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total \([<f l d\{, f l d\}\rangle]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{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 796) 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 \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function apply. This function supports the total qualifier in the same manner as described under the chi2test_p ([set_expression][ total [<fld \(\{, f l d\}>]\) col, row, observed_value [, expected_value]) (page 785) function.

\section*{Examples:}
```

ztestw_conf( Weight, Value-TestValue)

```

\section*{Special Input Field Aggregation Functions}

There are some special aggregation functions for input fields:
inputsum (inputfield[, distribution_mode][set_expression]) (page 798)
inputavg(inputfield [, distribution_mode][set_expression]) (page 798)

\section*{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.
' \(=\) ' \(\quad\) 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).

\section*{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.
' \(=\) ' \(\quad\) 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).

\section*{Examples:}
```

inputavg( Budget )
inputavg ( Budget, '+' )
inputavg ( Budget, '*' )
inputavg ( Budget, '=' )
inputavg ( Budget, '/' )
inputavg ( Budget, '+T' )
inputavg ( Budget, '+A' )

```

\section*{Set Analysis}

Sets can be used in aggregation functions. Aggregation functions normally aggregate over the set of possible records defined by the current selection. But an alternative set of records can be defined by a set expression. Hence, a set is conceptually similar to a selection.
A set expression always begins and ends with curly brackets when used, e.g. \{BM01\}.

\section*{Set Identifiers}

There is one constant that can be used to denote a record set; 1. It represents the full set of all the records in the application.
The \(\$\) sign represents the records of the current selection. The set expression \(\{\$\}\) is thus the equivalent of not stating a set expression. \(\{\mathbf{1}-\mathbf{\$}\}\) is all the more interesting as it defines the inverse of the current selection, i.e. everything that the current selection excludes.
Selections from the Back/Forward stack can be used as set identifiers, by use of the dollar symbol: \$1 represents the previous selection, i.e. equivalent to pressing the Back button. Similarly, \$_1 represents one step forward, i.e. equivalent to pressing the Forward button. Any unsigned integer can be used in the Back and Forward notations, i.e. \(\$ 0\) represents the current selection..
Finally, bookmarks can be used as set identifiers. Note that only server and document bookmarks can be used as set identifiers. Either the bookmark ID or the bookmark name can be used, e.g. BM01 or MyBookmark. Only the selection part of a bookmark is used. The values are not included. It is thus not possible to use input fields in bookmarks for set analysis.

\section*{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( \(\left\{\$ \_2\right\}\) Sales )
returns sales for the 2 nd 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({DocumentlMyBookmark}Sales)
returns the sales for the document bookmark MyBookmark.

```

\section*{Set Operators}

Several set operators that can be used in set expressions exist. All set operators use sets as operands, as described above, and return a set as result.
+ Union. This binary operation returns a set consisting of the records that belong to any of the two set operands.
- Exclusion. This binary operation returns a set of the records that belong 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.
The order of precedence is 1) Unary minus (complement), 2) Intersection and Symmetric difference, and 3) Union and Exclusion. Within a group, the expression is evaluated from left to right. Alternative orders can be defined by standard brackets, which may be necessary since the set operators do not commute, e.g. \(\mathrm{A}+(\mathrm{B}-\) \(C)\) is different from \((A+B)-C\) which in turn is different from (A-C)+B.

\section*{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.

```

\section*{Note!}

The use of set operators in combination with basic aggregation expressions involving fields from multiple QlikView tables may cause unpredictable results and should be avoided. E.g. if Quantity and Price are fields from different tables, then the expression sum( \(\{\$ * \mathrm{BM} 01\}\) Quantity*Price) should be avoided.

\section*{Set Modifiers}

A set can be modified by an additional or a changed selection. Such a modification can be written in the set expression. The modifier consists of one or several field names, each followed by a selection that should be made on the field, all enclosed by \(<\) and \(>\). E.g. \(<\) Year \(=\{2007,+2008\}\), Region \(=\{\) US \(\}>\). Field names and field values can be quoted as usual, e.g. <[Sales Region]=\{'West coast', 'South America'\}>.
There are several ways to define the selection: A simple case is a selection based on the selected values of another field, e.g. \(<\) OrderDate \(=\) DeliveryDate \(>\). This modifier will take the selected values from DeliveryDate and apply those as a selection on OrderDate. If there are many distinct values - more than a couple of hundred - then this operation is CPU intense and should be avoided.

The most common case, however, is a selection based on a field value list enclosed in curly brackets, the values separated by commas, e.g. \(<\) Year \(=\{2007,2008\}>\). The curly brackets here define an element set, where the elements can be either field values or searches of field values. A search is always defined by the use of double quotes, e.g. \(<\) Ingredient \(=\{" *\) Garlic*" \(\}>\) will select all ingredients including the string 'garlic'. Searches are case-insensitive and are made also over excluded values.
Empty element sets, either explicitly e.g. \(<\) Product \(=\{ \}>\) or implicitly e.g. \(<\) Product \(=\{\) "Perpetuum Mobile" \(\}>\) (a search with no hits) mean no product, i.e. they will result in a set of records that are not associated with any product. Note that this set cannot be achieved through usual selections, unless a selection is made in another field, e.g. TransactionID.
Finally, for fields in and-mode, there is also the possibility of forced exclusion. If you want to force exclusion of specific field values, you will need to use " \(\sim\) " in front of the field name.
A set modifier can be used on a set identifier or on its own. It cannot be used on a set expression. When used on a set identifier, the modifier must be written immediately after the set identifier, e.g. \(\{\$<\) Year \(=\) \(\{2007,2008\}>\}\). When used on its own, it is interpreted as a modification of the current selection.

\section*{Examples:}
\(\operatorname{sum}(\{1<\) Region \(=\{U S\}>\}\) Sales \()\)
returns the sales for region US disregarding the current selection.
\(\operatorname{sum}(\{\$<\) Region \(=>\}\) Sales \()\)
returns the sales for the current selection, but with the selection in "Region" removed.
\(\operatorname{sum}(\{<\) Region \(=>\}\) Sales \()\)
returns the same as the example immediately above. When the set to modify is omitted, \(\$\) is assumed.

\section*{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.
```

$\operatorname{sum}(\{\$<$ 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".
\(\operatorname{sum}(\{\$<\sim\) Ingredient \(=\{" *\) garlic*" \(\}>\}\) Sales \()\)
returns the sales for current selection, but with a forced exclusion of all Ingredients containing the string 'garlic'.
\(\operatorname{sum}(\{\$<\) 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( \(\left\{\$<\right.\) Year \(\left.=\left\{" 2 * ", " 198^{* "}\right\}>\right\}\) 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.

\section*{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".

\section*{Examples:}
\(\operatorname{sum}(\{\$<\) Product \(=\) Product \(+\{\) OurProduct 1\(\}-\{\) OurProduct 2\(\}>\}\) 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.
\(\operatorname{sum}(\{\$<\) 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'.

\section*{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 \(\langle\) Year \(=\) Year \(+\{2007,2008\}\rangle\). A short and equivalent way to write this is \(\langle\) Year \(+=\{2007\), \(2008\}>\), i.e. the assignment operator implicitly defines a union. Also implicit intersections, exclusions and symmetric differences can be defined using "*=", " \(=="\) and " \(="\).

\section*{Examples:}
sum \((\{\$<\) Product \(+=\{\) OurProduct1, OurProduct 2\(\}>\}\) Sales )
returns the sales for the current selection, but using an implicit union to add the products "Our-
Product1" 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 \(*=\{\) OurProduct 1\(\}>\}\) Sales \()\)
returns the sales for the current selection, but only for the intersection of currently selected products and the product "OurProduct1".

\section*{Set Modifiers with Dollar-Sign Expansions}

Variables and other dollar-sign expansions can be used in set expressions.

\section*{Examples:}
sum( \(\{\$<\) Year \(=\{\$(\# v L a s t Y e a r)\}>\}\) 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.

\section*{Set Modifiers with Advanced Searches}

Advanced searches using wildcards and aggregations can be used to define sets.

\section*{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.
\(\boldsymbol{\operatorname { s u m }}(\{\$<\) Customer \(=\{"=\operatorname{Sum}(\{1<\) 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 .

\section*{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. \(\mathrm{P}(\{1\}\) Customer). These functions cannot be used in other expressions:

\section*{Examples:}
\(\operatorname{sum}(\{\$<\) Customer \(=\mathrm{P}(\{1<\) 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.
\(\boldsymbol{\operatorname { s u m }}(\{\$<\) Customer \(=\mathrm{P}(\{1<\) 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.
\(\boldsymbol{\operatorname { s u m }}(\{\$<\) Customer \(=\mathrm{P}(\{1<\) 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.
\(\operatorname{sum}(\{\$<\) Customer \(=\mathrm{E}(\{1<\) Product \(=\{\) 'Shoe' \(\}>\})>\}\) Sales \()\)
returns the sales for current selection, but only those customers that never bought the product 'Shoe'. The element function \(\mathrm{E}(\mathrm{)}\) here returns the list of excluded customers; those that are excluded by the selection 'Shoe' in the field Product.

\section*{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 [=| += | ᄀ-= |*= |/= ] element_set_expression
element_set_expression ::= element_set { set_operator element_set }
element_set ::= [field_name ] | {lement_list }| element_function
element_list ::= element {, element }
element_function ::= (\mathbf{P}|\mathbf{E )([ set_expression ] [field_name ] )}
element ::= field_value | " search_mask"

```

\subsection*{68.5 Synthetic Dimension Functions}

\section*{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.

\section*{Examples:}
```

valuelist ( 1, 10, 100 )
valuelist ( 'a', 'xyz', 55 )

```

\section*{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.

\section*{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

\subsection*{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:
\begin{tabular}{lr} 
Chart General Numeric Functions & page 805 \\
Chart Range Functions & page 809 \\
Exponential and Logarithmic Functions & page 312 \\
Trigonometric and Hyperbolic Functions & page 313 \\
Statistical Distribution Functions & page 314 \\
Financial Functions & page 315 \\
Mathematical Constants and Parameter-Free Functions & page 318 \\
String Functions & page 320 \\
Chart Inter Record Functions & page 815 \\
Document Functions & page 337 \\
Chart Ranking Functions & page 821 \\
Conditional Functions & page 328 \\
Logical Functions & page 329 \\
NULL Functions & page 330 \\
System Functions & page 330 \\
Date and Time Functions & page 338 \\
Interpretation Functions & page 825 \\
Formatting Functions & page 356 \\
Color Functions & page 361
\end{tabular}

\section*{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.

\section*{Examples:}
\(\operatorname{div}(7,2)\) returns 3
\(\operatorname{div}(\mathbf{9 , 3})\) returns 3
\(\operatorname{div}(-4,3)\) returns -1
\(\operatorname{div}(4,-\mathbf{3})\) returns -1
\(\operatorname{div}(-4,-3)\) returns 1
```

mod}(x1,x2

```

Mathematical modula function. Both parameters must have integer values. \(x 2\) must be greater than 0 . The result is the non-negative remainder of an integer division.

\section*{Examples:}
\(\bmod (7,2)\) returns 1
\(\bmod (7.5,2)\) returns NULL
\(\bmod (\mathbf{9 , 3})\) returns 0
\(\bmod (-4,3)\) returns 2
\(\bmod (4,-3)\) returns NULL
\(\bmod (-4,-3)\) returns NULL

\section*{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.

\section*{Examples:}
fmod (7,2) returns 1
\(\operatorname{fmod}(7.5,2)\) returns 1.5
fmod \(\mathbf{9 , 3}\) ) returns 0
fmod \((-4,3)\) returns -1
fmod ( 4,-3) returns 1
\(\mathbf{f m o d}(-4,-3)\) 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.

\section*{Examples:}
\[
\text { ceil( } 2.4 \text { ) returns } 3
\]
ceil( 2.6 ) returns 3
ceil( \(\mathbf{3 . 8 8}, \mathbf{0 . 1}\) ) returns 3.9
\(\operatorname{ceil}(3.88,5)\) returns 5
\(\operatorname{ceil}(1.1,1,0.5)\) returns 1.5

\section*{floor (x [ , base [ , offset ] ])}

Rounding of \(x\) downwards to the nearest multiple of base with an offset of offset. The result is a number.

\section*{Examples:}
floor( 2.4 ) returns 2
floor( 2.6 ) returns 2
floor ( \(\mathbf{3 . 8 8}, \mathbf{0 . 1}\) ) returns 3.8
floor( \(3.88,5\) ) returns 0
floor( \(\mathbf{1 . 1}, \mathbf{1}, \mathbf{0 . 5}\) ) returns 0.5

\section*{frac (x)}

Returns the fraction part of \(x\). The fraction is defined in such a way that \(\operatorname{frac}(x)+\boldsymbol{f l o o r}(x)=x\).

\section*{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.

\section*{Examples:}
round( 2.4 ) returns 2
round( 2.6 ) returns 3
round( 2.5 ) returns 3
round( \(\mathbf{3 . 8 8}, \mathbf{0 . 1}\) ) returns 3.9
round ( \(\mathbf{3 . 8 8}, 5\) ) returns 5
round( \(1.1, \mathbf{1}, 0.5\) ) returns 1.5

\section*{fabs (x)}

The absolute value of \(x\). The result is a positive number.

\section*{Examples:}
fabs( 2.4 ) returns 2.4
fabs( - \(\mathbf{- 3 . 8}\) ) 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 nonnumeric values as 0 .

\section*{Examples:}
numsum( \(1,2,4\) ) returns 7
numsum( \(1,{ }^{\prime} \times \mathbf{x y z} \mathbf{'}^{\prime}\) ) returns 1
```

numsum( null() ) returns 0

```

\section*{Note!}

The numsum function is now obsolete and is superseded by the rangesum(exprl [, expr2, \(\ldots \operatorname{expr} N]\) ) (page 809) function.

\section*{numcount (expr1 [ , expr2, ... exprN ])}

Returns the number of numeric values found in 1 to N arguments

\section*{Examples:}
numcount( \(\mathbf{1 , 2 , 4 ,}\) ) returns 3
numcount ( \(\mathbf{2 , x y z}\) ) returns 1
numcount( null( ) ) returns 0

\section*{Note!}

The numcount function is now obsolete and is superseded by the rangecount (exprl [, expr2, \(\ldots \operatorname{expr} N])\) (page 810) function.

\section*{numavg (expr1 [ , expr2, ... exprN ])}

Returns the numeric average of 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
numavg( \(\mathbf{1 , 2 , 4}\) ) returns 2.33333333
numavg( \(\mathbf{1 , ' x y z} \mathbf{'}^{\prime}\) ) returns 1
numavg( null( ) 'abc' ) returns NULL

\section*{Note!}

The numavg function is now obsolete and is superseded by the rangeavg(expr1 [, expr2, ... exprN ]) (page 809) function.
```

nummin(expr1 [ , expr2, ... exprN ])

```

Returns the lowest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
nummin( \(1,2,4\) ) returns 1
nummin( 1, 'xyz' \(^{\prime}\) ) returns 1
nummin( null( ) 'abc' ) returns NULL

\section*{Note!}

The nummin function is now obsolete and is superseded by the rangemin(exprl [, expr2, \(\ldots\) exprN ]) (page 810) function.
```

nummax(expr1 [ , expr2, ... exprN ])

```

Returns the highest numeric value of 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
```

nummax ( 1,2,4 ) returns 4

```
nummax ( 1, 'xyz' ) returns 1
nummax( null() 'abc' ) returns NULL

\section*{Note!}

The nummax function is now obsolete and is superseded by the rangemax (expr1 [, expr2, ... exprN ]) (page 810) function.

\section*{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.

\section*{Examples:}
fact( 1 ) returns 1
fact( 5 ) returns \(120(1 * 2 * 3 * 4 * 5=120)\)
fact( \(\mathbf{- 5}\) ) returns NULL

\section*{combin (n1, n2)}

Returns the number of combinations of \(n 2\) items that can be picked from a group of \(n 1\) items. The order in which the items are selected is insignificant. Non-integer items will be truncated.

\section*{Example:}

How many combinations of 7 numbers can be picked from a total of 35 Lotto numbers?
combin( 35,7 ) returns 6724520
```

permut(n1, n2)

```

Returns the number of permutations of \(n 2\) items that can be selected from a group of \(n 1\) items. The order in which the items are selected is significant. Non-integer arguments will be truncated.

\section*{Example:}

In how many ways could the gold, silver and bronze medals be distributed after a 100 m final with 8 participants?
permut( 8,3 ) returns 336

\section*{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.

\section*{Examples:}
even( 3 ) returns false
\(\operatorname{even}(2 * 10)\) returns true
\(\operatorname{even}(3.14)\) returns NULL

\section*{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.

\section*{Examples:}
odd( 3 ) returns true
\(\operatorname{odd}(2 * 10)\) returns false
odd( 3.14 ) returns NULL

\section*{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.

\section*{Examples:}
\(\operatorname{sign}(66)\) returns 1
\(\boldsymbol{\operatorname { s i g n }}(0)\) returns 0
\(\boldsymbol{\operatorname { s i g n }}(\mathbf{- 2 3 4})\) returns -1

\section*{bitcount(i)}

Returns the number of set bits in \(i\). Where \(i\) is interpreted as a signed 32-bit integer.

\section*{Examples:}
bitcount ( 3 ) returns 2
bitcount ( \(\mathbf{- 1}\) ) returns 32
bitcount ( \(\boldsymbol{n}\) )+bitcount (bitnot \(\boldsymbol{n}\) ) returns 32 (as long as \(n\) is a valid integer)

\section*{Chart Range Functions}

Range functions will replace the following Chart General Numeric Functions (page 805): 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 nonnumeric values as 0 .
The argument expressions of this function may contain Chart Inter Record Functions (page 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangesum (1,2,4) returns 7
rangesum (1,'xyz') returns 1
rangesum (null()) returns 0
rangesum
(above(count(x),-1,3))

```
returns the sum of the three results of the \(\operatorname{count}(\mathbf{x})\) function evaluated on the row below the current row, the current row and the row above the current row.
```

rangeavg(expr1 [ , expr2, ... exprN ])

```

Returns the average of a range of 1 to N arguments. If no numeric value is found, NULL is returned.
The argument expressions of this function may contain Chart Inter Record Functions (page 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangeavg(1,2,4) returns 2.33333333
rangeavg (1,'xyz') returns 1
rangeavg (null(), 'abc') returns NULL
rangeavg
(above(count(x),0,3))
returns a sliding average of the result of the inner count(\mathbf{x) expression cal-}
culated 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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangecount (1,2,4) returns 3
rangecount (2,'xyz') returns 2
rangecount (null( )) returns 0
rangecount
(above(sum(x),-1,3))
returns the number of values within the three results of the $\operatorname{sum}(\mathbf{x})$ function evaluated on the row below the current row, the current row and the row above the current row.

```
```

rangemin(expr1 [ , expr2, ... exprN ])

```

Returns the lowest numeric value found within a range 1 to N arguments. If no numeric value is found, NULL is returned.

\section*{Examples:}
```

rangemin (1,2,4) returns 1
rangemin (1,'xyz') returns 1
rangemin (null( ), returns NULL

```
'abc')
rangemin returns the lowest of the three results of the \(\boldsymbol{\operatorname { s u m } ( x )}\) function evaluated on the
(above(sum(x),-1,3)) 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.

\section*{Examples:}
```

rangemax (1,2,4) returns 4
rangemax (1,'xyz') returns 1
rangemax (null( ), returns NULL
'abc')

```
rangemax returns the highest of the three results of the \(\operatorname{sum}(x)\) function evaluated on the (above(sum(x),-1,3)) 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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
rangestdev (1,2,4)
rangestdev (null( ))
rangestdev
(above(count(x),0,3))
returns 1.5275252316519
returns NULL
returns a sliding standard deviation of the result of the inner count(x) expression calculated on the current row and two rows above the current row.
```

rangeskew(expr1 [ , expr2, ... exprN ])

```

Returns the skewness within a range of 1 to N arguments. If no numeric value is found, NULL is returned. The argument expressions of this function may contain Chart Inter Record Functions (page 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangeskew (1,2,4)
returns 0.93521952958283

```
rangeskew
(above(count(x),0,3))
returns a sliding skewness of the result of the inner \(\boldsymbol{c o u n t}(\mathbf{x})\) expression calculated on the current row and two rows above the current row.
```

rangekurtosis(expr1 [ , expr2, ... exprN ])

```

Returns the kurtosis within a range of 1 to N arguments. If no numeric value is found, NULL is returned. The argument expressions of this function may contain Chart Inter Record Functions (page 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangekurtosis (1,2,4,7) returns -0.28571428571429
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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangefractile returns 1
(0.24,1,2,4,6)
rangefractile returns 3
(0.5,1,2,3,4,6)
rangefractile returns 3.5
(0.5,1,2,5,6)
rangefractile (0.5, returns the median of the three results of the \boldsymbol{sum}(\mathbf{x})\mathrm{ function evaluated on the}
above(sum(x),-1,3))
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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangenumericcount returns 3
(1,2,4)
rangenumericcount returns 1
(2,'xyz')
rangenumericcount
returns 0
(null())
rangenumericcount
(above(sum(x),-1,3))
returns the number of numeric values within the three results of the $\operatorname{sum}(\mathbf{x})$ 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 815) with a third optional parameter, which in themselves return a range of values.

\section*{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 \(\operatorname{sum}(\mathbf{x})\) 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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangenullcount (1,2,4) returns 0
rangenullcount returns 0
(2,'xyz')
rangenullcount (null( returns 2
),null( ))
rangenullcount
(above(sum(x),-1,3))
returns the number of NULL values within the three results of the $\operatorname{sum}(\mathbf{x})$ function evaluated on the row below the current row, the current row and the row above the current row.

```

\section*{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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
\begin{tabular}{lr}
\begin{tabular}{ll} 
rangemissingcount \\
\((\mathbf{1 , 2 , 4})\)
\end{tabular} & returns 0 \\
rangemissingcount & returns 1 \\
\((\mathbf{2 , \prime x y z}\) ') & \\
rangemissingcount & returns 1
\end{tabular}
(null( ))
rangemissingcount
(above(sum(x),-1,3))
returns the number of non-numeric values within the three results of the \(\operatorname{sum}(\mathbf{x})\) function evaluated on the row below the current row, the current row and the row above the current row.

\section*{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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangeminstring(1,2,4) returns 1
rangeminstring returns 'abc'
('xyz','abc')
rangeminstring (null()) returns NULL
rangeminstring
(above(sum(x),-1,3))

```
returns 1
returns 'abc'
returns NULL
returns the first (in text sort order) of the three results of the \(\operatorname{sum}(\mathbf{x})\) 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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangemaxstring(1,2,4) returns 4
rangemaxstring returns 'xyz'
('xyz','abc')
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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangemode (1,2,9,2,4) returns 2

```
rangemode ('a',4,' \(\mathbf{a}^{\prime}, \mathbf{4}\) ) returns NULL
rangemode (null( )) returns NULL
rangemode returns the most commonly occurring value within the three results of the
(above(sum(x),-1,3)) \(\operatorname{sum}(\mathbf{x})\) 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 815) with a third optional parameter, which in themselves return a range of values.

\section*{Examples:}
```

rangeonly (1,2,4) returns NULL
rangeonly (1,'xyz') returns NULL
rangeonly (null( ), returns 'abc'
'abc')
rangeonly
(above(sum(x),-1,3))
uses the three results of the $\operatorname{sum}(\mathbf{x})$ function evaluated on the row below the current row, the current row and the row above the current row. If they contain exactly one value, that value will be returned.

```
```

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 815) 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.

\section*{Examples:}
rangecorrel ( \(\mathbf{( 2 , 3 , 6 , 8 , 9 , 4}\) ) returns 0,269
rangecorrel (above(sum(X), \(\mathbf{0}, 10)\), above( \(\mathbf{Y}, \mathbf{0}, 10)\) )
rangecorrel (above(total \(\mathrm{X}, 0\), rowno(total)), above(total Y,0,rowno(total)))

\section*{Chart Inter Record Functions}

These functions can only be used in chart expressions.

\begin{abstract}
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.
\end{abstract}

Suppression of zero values is automatically disabled when these functions are used.
See Examples of Chart Inter Record Functions (page 839).

\section*{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.

\section*{Example:}
```

fieldvalue( 'Helptext', 5 )

```

\section*{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.

\section*{Example:}
```

fieldindex( 'Name', 'John Doe' )

```

\section*{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 839).

\section*{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 839).
```

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 809).

\section*{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.

\section*{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 809).

\section*{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.

\section*{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 809).

\section*{Examples:}
```

sum( Sales ) / top( sum( Sales ))
top( sum( Sales ), 2 )
top( total sum( Sales ))

```
rangeavg returns an average of the results of the \(\operatorname{sum}(x)\) function evaluated on the five \((\boldsymbol{t o p}(\operatorname{sum}(\mathbf{x}), \mathbf{1 , 5}))\) 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 809).

\section*{Examples:}
```

bottom( sum( Sales ))
bottom( sum( Sales ), 2 )
bottom( total sum( Sales ))

```
rangeavg (bot- returns an average of the results of the \(\operatorname{sum}(x)\) function evaluated on the five
\(\operatorname{tom}(\operatorname{sum}(\mathbf{x}), \mathbf{1}, \mathbf{5}))\) bottom rows of the current column segment.

\section*{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 809).

\section*{Examples:}
```

before( sum( Sales ))
before( sum( Sales ), 2 )
before( total sum( Sales ))

```
\begin{tabular}{ll} 
rangeavg & returns an average of the three results of the \(\operatorname{sum}(x)\) function evaluated in the three columns \\
\(\boldsymbol{( b e f o r e}(\operatorname{sum}(\mathbf{x}), \mathbf{1 , 3}))\) & immediately to the left of the current column.
\end{tabular}

\section*{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 809).

\section*{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 \((\operatorname{after}(\operatorname{sum}(\mathbf{x}), \mathbf{1 , 3})) \quad\) three columns immediately to the right of the current column.

\section*{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 809).

\section*{Examples:}
```

first( sum( Sales ))
first( sum( Sales ), 2 )
first( total sum( Sales )

```
```

rangeavg
(first(sum(x),1,5))
returns an average of the results of the sum(x) function evaluated on the five leftmost columns of the current row segment.

```

\section*{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 809).

\section*{Examples:}
```

last( sum( Sales ))
last( sum( Sales ), 2 )
last( total sum( Sales )

```
rangeavg
(last(sum(x),1,5))
returns an average of the results of the \(\operatorname{sum}(\mathrm{x})\) function evaluated on the five rightmost columns of the current row segment.

\section*{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.

\section*{Example:}
```

if( RowNo( )=1, 0, sum( Sales ) / above( sum( Sales )))

```

\section*{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.

\section*{Example:}
```

if( ColumnNo( )=1, 0, sum( Sales ) / before( sum( Sales )))

```

\section*{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.

\section*{Example:}
```

if( RowNo( )=NoOfRows( ), 0, after( sum( Sales )))

```

\section*{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.

\section*{Example:}
if ( ColumnNo( )=NoOfColumns( ), 0, after ( sum( Sales )))

\section*{fieldvaluecount(fieldname )}

Returns the number of distinct values in a field. Fieldname must be given as a string (e.g. a quoted literal).

\section*{Column (ColumnNo)}

Returns the value found in the column ColumnNo in a straight table or a pivot table.

\section*{Example:}
column (1)/column (2) returns the quotient.

\section*{Chart Ranking Functions}

These functions can only be used in chart expressions.

\section*{Note!}

Suppression of zero values is automatically disabled when these functions are used. NULL values are disregarded.

\section*{Examples of Chart Ranking Functions (page 835)}
```

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.

\section*{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.

\section*{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
0 (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.

1
Lowest rank on all columns in the group.

2
Average rank on all columns in the group.

3
Highest rank on all columns in the group.
4
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.

\section*{format}

0 (default)
1 Low value on all columns in the group.
2

The order of columns for mode 4 and format 2 is determined by the sort order of the chart dimensions.

\section*{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.

\section*{Chart Field Functions}

The field functions are:

\section*{getcurrentfield (GroupName)}

Returns the name of the currently active field in a group.

\section*{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 .

\section*{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 \(\langle\mathrm{CR}><\mathrm{LF}>\).
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 .

\section*{Examples}
```

getcurrentselections ( )
getcurrentselections ( chr(13)\&chr(10) , ' = ' )
getcurrentselections ( chr(13)\&chr(10) , ' : ' , ' ; ' )
getcurrentselections ( chr(13)\&chr(10) , ': ' , ' ; ' , 10 )

```

\section*{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.

\section*{Examples}
```

    getselectedcount ( Year )
    getselectedcount ( Year, true( ) )

```

\section*{getnotselectedcount ( FieldName [, IncludeExcluded])}

Returns the current number of not-selected values in a field in a field with And-Mode in List Boxes (page 117) (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.

\section*{Examples}
```

getnotselectedcount ( Year )
getnotselectedcount (Year,true())

```

\section*{getpossiblecount (FieldName)}

Returns the current number of possible values in a field.

\section*{Example}
```

    getpossiblecount ( Year )
    ```
```

getexcludedcount (FieldName)

```

Returns the current number of excluded values in a field.

\section*{Example}
```

getexcludedcount ( Year )

```
```

getalternativecount (FieldName)

```

Returns the current number of alternative values in a field.

\section*{Example}
```

getalternativecount ( Year )

```

\section*{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.

\section*{Note!}

For reasons of clarity, all number representations are given with decimal point as decimal separator.
The following interpretation functions are available:
\begin{tabular}{ll} 
Num\# & page 826 \\
Money\# & page 827 \\
Date\# & page 827
\end{tabular}
\begin{tabular}{lr} 
Time\# & page 828 \\
Timestamp\# & page 828 \\
Interval\# & page 829 \\
Text & page 829 \\
Num\# & \\
num\# (expression [ , format-code [ , decimal-sep [, thousands-sep] ] \\
])
\end{tabular}

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.

\section*{Examples:}

The examples below assume the two following operating system settings:
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|r|}{Default setting 1} \\
\hline \multicolumn{4}{|l|}{Number format} \\
\hline \multicolumn{4}{|r|}{num\#( \(\mathbf{A}\), '\#' ) where \(A=35,6\)} \\
\hline & Settin & & Setting 2 \\
\hline String & 35,64 & . 375 & 35648.375 \\
\hline Number & & & 35648.375 \\
\hline
\end{tabular}
num\#( A, '\#.\#', ' \({ }^{\prime}\), ',') where \(A=35,648.375\) returns:
Setting 1 Setting 2
String \(\quad 35,648.375 \quad 35,648.375\)
Number \(35648.375 \quad 35648.375\)
num\#( A, '\#.\#',',','.' ) where A=35648.375 returns:
Setting 1 Setting 2
String \(\quad 35648.375 \quad 35648.375\)
Number 3564837535648375
\begin{tabular}{lrr}
\multicolumn{3}{c}{ num\#( A, 'abc\#,\#' \()\) wher } \\
& Setting 1 & Setting 2 \\
String & abc123,4 & abc123,4 \\
Number & 123.4 & 1234
\end{tabular}

\section*{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.

\section*{Examples:}

The examples below assume the two following operating system settings:
\begin{tabular}{ccc} 
& Default setting 1 & Default setting 2 \\
Money format & \(\mathrm{kr} \# \# \# 0,00\) & \$\#,\#\#0.00
\end{tabular}
money\#( , '\# \#\#0,00 kr' ) where A=35 648,37 kr returns:
\begin{tabular}{lrr} 
& \multicolumn{1}{l}{ Setting 1 } & \multicolumn{1}{l}{ Setting 2 } \\
String & 35648.37 kr & 35648.37 kr \\
Number & 35648.37 & 3564837
\end{tabular}
money\#( A, ' \$\#', '.', ',' ) where \(A=\$ 35,648.37\) returns:
\begin{tabular}{lrr} 
& \multicolumn{1}{l}{ Setting 1 } & \multicolumn{1}{l}{ Setting 2 } \\
String & \(\$ 35,648.37\) & \(\$ 35,648.37\) \\
Number & 35648.37 & 35648.37 \\
Date\# & & \\
Dat &
\end{tabular}

\section*{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.

\section*{Examples:}

The examples below assume the two following operating system settings:
Default setting 1 Default setting 2
Date format \(\quad\) YY-MM-DD \(\quad\) M/D/YY
date\# ( A ) where A=8/6/97 returns:
Setting 1 Setting 2
\begin{tabular}{lcc} 
String & \(8 / 6 / 97\) & \(8 / 6 / 97\) \\
Number & - & 35648
\end{tabular}
date\# ( A, 'YYYY.MM.DD' ) where A=1997.08.06 returns:

\section*{Setting 1}

Setting 2
\begin{tabular}{lll} 
String & 1997.08 .06 & 1997.08 .06 \\
Number & 35648 & 35648
\end{tabular}

\section*{Time\#}

\section*{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.

\section*{Examples:}

The examples below assume the two following operating system settings:
\begin{tabular}{ccc} 
& Default setting 1 & Default setting 2 \\
Time format & hh:mm:ss & hh.mm.ss
\end{tabular}
time\#( A ) where \(\mathrm{A}=09: 00: 00\) returns:

\section*{Setting 1 Setting 2}
\begin{tabular}{lrc} 
String & \(09: 00: 00\) & \(09: 00: 00\) \\
Number & 0.375 & -
\end{tabular}
time\#( A, 'hh.mm' ) where A=09.00 returns:
Setting 1 Setting 2
\begin{tabular}{lll} 
String & 09.00 & 09.00 \\
Number & 0.375 & 0.375
\end{tabular}

\section*{Timestamp\#}

\section*{timestamp\#(expression [ , format-code ])}

The timestamp\# function evaluates the expression as a date and time according to the string given as formatcode. If the format code is omitted, the default date and time formats set in the operating system are used.

\section*{Examples:}

The examples below assume the two following operating system settings:
Default setting 1 Default setting 2
\begin{tabular}{lcc} 
Date format & YY-MM-DD & M/D/YY \\
Time format & hh:mm:ss & hh:mm:ss
\end{tabular}
timestamp\#( A ) where \(A=8 / 6 / 97\) 09:00:00 returns:
\begin{tabular}{llr} 
& \multicolumn{1}{l}{ Setting 1 } & \multicolumn{1}{l}{ Setting 2 } \\
String & \(8 / 6 / 97\) & \(09: 00: 00\)
\end{tabular} 8/6/97 09:00:00
timestamp\#( A, 'YYYY-MM-DD hh_mm' ) where A=8/6/97 09_00 returns:
Setting 1 Setting 2
String 1997-08-06 09_00 1997-08-06 09_00
\(\begin{array}{lll}\text { Number } & 35648.375 & 35648.375\end{array}\)

\section*{Interval\#}

\section*{interval\#(expression [ , format-code ])}

The interval\# function evaluates the expression as a time interval according to the string given as a formatcode. 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.

\section*{Examples:}

The examples below assume the following operating system settings:
\begin{tabular}{ll} 
Short date format: & YY-MM-DD \\
Time format: & \(\mathbf{h h}: \mathbf{m m}: \mathbf{s s}\) \\
Number decimal separator: & .
\end{tabular}
interval\#( A, 'D hh:mm' ) where A=1 09:00 returns:

String \(\quad 109: 00\)
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

\section*{Text}

\section*{text(expression )}

The text function forces the expression to be treated as text, even if a numeric interpretation is possible.

\section*{Examples:}
\(\operatorname{text}(\mathbf{A})\) where \(\mathrm{A}=1234\) returns:
Result
String 1234
Number
text( \(\mathbf{p i}())\) returns:

\section*{Result}

String \(\quad 3.1415926535898\)
Number

\section*{69 Examples}

\subsection*{69.1 Examples of Aggregate Qualifiers}

These examples are made with the sum([\{set_expression \(\}][d i s t i n c t][t o t a l[<f l d ~\{, f l d\}>]]\) expression) (page 771) function, but may be applied on all chart aggregation functions supporting Set Analysis (page 799) definitions and the total qualifier.

\section*{Example 1:}

Study the representation of a straight table below, depicting the logical state before any selections have been made:
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{6}{|c|}{ Aggregation function with total qualifier } \\
\hline \multicolumn{7}{|c|}{ Month } & Group & Sum(Value) & sum( total Value) & sum(\{1\} total Value) \\
\hline & & \(\mathbf{2 1}\) & \(\mathbf{2 1}\) & \(\mathbf{2 1}\) \\
\hline 1 & A & 1 & 21 & 21 \\
\hline 1 & B & 2 & 21 & 21 \\
\hline 2 & A & 3 & 21 & 21 \\
\hline 2 & B & 4 & 21 & 21 \\
\hline 3 & A & 5 & 21 & 21 \\
\hline 3 & B & 6 & 21 & 21 \\
\hline
\end{tabular}

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
\begin{tabular}{|c|c|c|c|c|}
\hline Month & Group & Sum(Value) & sum( total Value) & sum(\{1\} total Value) \\
\hline & & \(\mathbf{1 0}\) & \(\mathbf{1 0}\) & \(\mathbf{2 1}\) \\
\hline 1 & A & 1 & 10 & 21 \\
\hline 1 & B & 2 & 10 & 21 \\
\hline 2 & A & 3 & 10 & 21 \\
\hline 2 & B & 4 & 10 & 21 \\
\hline
\end{tabular}

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).

\section*{Example 2:}

Study the representation of a straight table below:
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{6}{|c|}{ total qualifier with listed fields } & & \\
\hline \multicolumn{2}{|c|}{ Month } & Group & Sum(Value) & sum( total <Month> Value)
\end{tabular} sum( total <Group> Value)

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.

\subsection*{69.2 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'

\section*{Synchronizing Selections between States}

\section*{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.

\section*{Example:}
\(\operatorname{count}(\{\) State \(1<\) Year \(=\$::\) Year, Month \(=\$::\) Month \(>\}\) DISTINCT [Invoice Number])
\(\operatorname{count}(\{\) State \(2<\) 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.

\section*{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.

\section*{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.

```

\begin{abstract}
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.
\end{abstract}

\section*{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.

\section*{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 Statel.
It is almost, but not quite equivalent to the following expression:
$\operatorname{count}(\{\$<$ 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 State 1 are passed into the default state. In the second expression, the selected values from State 1 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.

\section*{Examples:}
\(\boldsymbol{\operatorname { c o u n t }}(\{\$<[\) Invoice Number \(]=\mathrm{p}(\{\) State1 \(\}[\) Invoice Number \(]) * \mathrm{p}(\{\) State 2\(\}\) [Invoice Number] \()>\}\) DISTINCT [Invoice Number])
The expression to find the intersection of invoice numbers in the \(<\) default \(>\) state and State 1 is:
\(\operatorname{count}(\{\$<[\) Invoice Number] \(=\mathrm{p}(\{\$\}\) [Invoice Number]) \(* \mathrm{p}(\{\) State 1\(\}\) [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:

\section*{Examples:}
\(\operatorname{count}(\{\$<[\) Invoice Number \(]=p(\{\$\}[\) Invoice Number \(]) * p(\{\) State \(1<\) 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.

\subsection*{70.1 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.

\section*{Example 1:}

Study the two one-dimensional straight tables below:
\begin{tabular}{|l|r|r|}
\hline \multicolumn{3}{|l|}{ Single dimension RANK } \\
\hline Month & sum(Val) & rank(sum(Val)) \\
\hline & 110 & - \\
\hline 1 & 17 & 3 \\
\hline 2 & 9 & 7 \\
\hline 3 & 22 & 1 \\
\hline 4 & 16 & 4 \\
\hline 5 & 10 & 6 \\
\hline 6 & 5 & 8 \\
\hline 7 & 11 & 5 \\
\hline 8 & 20 & 2 \\
\hline \hline
\end{tabular}
\begin{tabular}{|l|r|r|}
\hline \multicolumn{3}{|l|}{ Sorted by ranking column } \\
\hline Month & sum(Val) & rank(sum(Val)) \\
\hline & 110 & - \\
\hline 3 & 22 & 1 \\
\hline 8 & 20 & 2 \\
\hline 1 & 17 & 3 \\
\hline 4 & 16 & 4 \\
\hline 7 & 11 & 5 \\
\hline 5 & 10 & 6 \\
\hline 2 & 9 & 7 \\
\hline 6 & 5 & 8 \\
\hline \hline
\end{tabular}

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.

\section*{Example 2:}

Study the two-dimensional pivot table below:
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{4}{|l|}{Two dimension RANK and the effect of tota} & \multirow[t]{2}{*}{} \\
\hline \multirow[t]{2}{*}{\begin{tabular}{|l|}
\hline Group \\
\hline A \\
\hline
\end{tabular}} & Month & sum(Val) & rank(sum(Val) & \\
\hline & -1 & 17 & 2 & 3 \\
\hline & 4 & 16 & 3 & 4 \\
\hline & 7 & 11 & 4 & 5 \\
\hline & 8 & 20 & 1 & 2 \\
\hline & Total & 64 & - & \\
\hline \multirow[t]{5}{*}{B} & \(\square 2\) & 9 & 3 & 7 \\
\hline & 3 & 22 & 1 & 1 \\
\hline & 5 & 10 & 2 & 6 \\
\hline & 6 & 5 & 4 & 8 \\
\hline & Total & 46 & - & - \\
\hline Total & & 110 & - & - \\
\hline
\end{tabular}

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.

\section*{Example 3:}

This example will demonstrate the effect of the different modes for the numeric representation of the ranking. Study the table below:
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline Month & \(\operatorname{sum}(\mathrm{X})\) & \(\operatorname{rank}(\mathrm{sum}(\mathrm{X})\) ) & mode=0 & mode=1 & mode=2 & mode=3 & mode=4 \\
\hline & 86 & - & - & - & - & - & \\
\hline 4 & 20 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline 2 & 12 & 2-3 & 2 & 2 & 2.5 & 3 & 2 \\
\hline 7 & 12 & 2-3 & 2 & 2 & 2.5 & 3 & 3 \\
\hline 3 & 10 & 4-5 & 4.5 & 4 & 4.5 & 5 & 5 \\
\hline 5 & 10 & 4-5 & 4.5 & 4 & 4.5 & 5 & 4 \\
\hline 8 & 9 & 6 & 6 & 6 & 6 & 6 & 6 \\
\hline 6 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
\hline 1 & 6 & 8 & 8 & 8 & 8 & 8 & 8 \\
\hline
\end{tabular}

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( \(\operatorname{rank}(\operatorname{sum}(X)\), 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.

\section*{Example 4:}

This example will demonstrate the effect of the different formats for the text representation of the ranking function. Study the table below:
\begin{tabular}{|l|r|r|r|r|}
\hline \multicolumn{5}{|c|}{ The effect of the format parameter on the result text representatio)- } \\
\hline Month & \(\operatorname{sum}(\mathrm{X})\) & \(\operatorname{rank}(\) sum \((X), 0,0)\) & \(\operatorname{rank}(\operatorname{sum}(X), 0,1)\) & \(\operatorname{rank}(\) sum \((X), 0,2)\) \\
\hline & 86 & - & - & -1 \\
\hline 4 & 20 & 1 & 1 & 1 \\
\hline 2 & 12 & \(2-3\) & 2 & 2 \\
\hline 7 & 12 & \(2-3\) & 2 & \\
\hline 3 & 10 & \(4-5\) & 4 & \\
\hline 5 & 10 & \(4-5\) & 4 & 4 \\
\hline 8 & 9 & 6 & 6 & 6 \\
\hline 6 & 7 & 7 & 7 & 7 \\
\hline 1 & 6 & 8 & 8 & 8 \\
\hline \hline
\end{tabular}

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.

\subsection*{71.1 Examples of Chart Inter Record Functions}

\section*{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.

\section*{Example 1:}

Study the straight table below, depicting the use of the top function in a one-dimensional table:
\begin{tabular}{|l|r|rr|}
\hline \multicolumn{4}{|c|}{ Single dimension straight table - top } \\
\hline Month & sum(Val) & top(sum(Val) & sum(Val) fop(sum(Val)) \\
\hline & 21 & 3 & \(700 \%\) \\
\hline 1 & 3 & 3 & \(100 \%\) \\
\hline 2 & 7 & 3 & \(233 \%\) \\
\hline 3 & 11 & 3 & \(367 \%\) \\
\hline \hline
\end{tabular}

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.

\section*{Example 2:}

Below is a two-dimensional straight table sorted primarily on the field Grp.
\begin{tabular}{|l|l|r|r|r|}
\hline \multicolumn{5}{|c|}{ Two dimension straight table - efifect of total qualifier } \\
\hline Month & Grp & sum(Val) & top(sum(Val) \()\) & top(total sum(Val)) \\
\hline & & 21 & - & 1 \\
\hline 1 & A & 1 & 1 & 1 \\
\hline 2 & A & 3 & 1 & 1 \\
\hline 3 & A & 5 & 1 & 1 \\
\hline 1 & B & 2 & 2 & 1 \\
\hline 2 & B & 4 & 2 & 1 \\
\hline 3 & B & 6 & 2 & 1 \\
\hline \hline
\end{tabular}

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 \(\mathrm{Grp}=\mathrm{A}\) and one for Grp \(=\mathrm{B}\).
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 he 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.
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{Two dimension pivot table - effect of total qualifier -} \\
\hline Month & Grp & sum(Val) & top(sum(Val) & top(total sum(Val)) \\
\hline \multirow[t]{3}{*}{1 E} & A & 1 & , & 1 \\
\hline & B & 2 & 1 & 1 \\
\hline & Total & 3 & 1 & \\
\hline \multirow[t]{3}{*}{2} & A & 3 & 3 & 1 \\
\hline & B & 4 & 3 & 1 \\
\hline & Total & 7 & 3 & \\
\hline \multirow[t]{3}{*}{3} & A & 5 & 5 & 1 \\
\hline & B & 6 & 5 & 1 \\
\hline & Total & 11 & 5 & \\
\hline Total & & 21 & - & 1 \\
\hline
\end{tabular}

The expression using the top function without he 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.

\section*{Example 3:}

Study the following straight table sorted on the field Grp:
\begin{tabular}{|l|l|l|l|l|}
\hline \multicolumn{5}{|c|}{ Two dimension straight table - sort on Grp } \\
\hline Month & Grp & sum(Val) & top(sum(Val) \()\) & sum(Val)Atop(sum(Val)) \\
\hline & & 21 & - & \\
\hline 1 & A & 1 & 1 & \(100 \%\) \\
\hline 2 & A & 3 & 1 & \(300 \%\) \\
\hline 3 & A & 5 & 1 & \(500 \%\) \\
\hline 1 & B & 2 & 2 & \(100 \%\) \\
\hline 2 & B & 4 & 2 & \(200 \%\) \\
\hline 3 & B & 6 & 2 & \(300 \%\) \\
\hline \hline
\end{tabular}

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:
\begin{tabular}{|l|l|r|r|r|}
\hline \multicolumn{5}{|c|}{ Two dimension straight table - sort on Month } \\
\hline Month & Grp & sum(Val) & top(sum(Val)) & sum(Val)/top(sum(Val)) \\
\hline & & 21 & - & - \\
\hline 1 & A & 1 & 1 & \(100 \%\) \\
\hline 1 & B & 2 & 1 & \(200 \%\) \\
\hline 2 & A & 3 & 3 & \(100 \%\) \\
\hline 2 & B & 4 & 3 & \(133 \%\) \\
\hline 3 & A & 5 & 5 & \(100 \%\) \\
\hline 3 & B & 6 & 5 & \(120 \%\) \\
\hline \hline
\end{tabular}

\section*{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.

\section*{Example 1:}

Study the straight table below, depicting the use of the above function in a one-dimensional table:
\begin{tabular}{|l|r|r|r|}
\hline \multicolumn{4}{|c|}{ Single dimension straight table - above } \\
\hline Month & sum(Val) & above(sum(Val) \()\) & sum(Val)/above(sum(Val)) \\
\hline & 21 & - & - \\
\hline 1 & 3 & - & - \\
\hline 2 & 7 & 3 & \(233 \%\) \\
\hline 3 & 11 & 7 & \(157 \%\) \\
\hline \hline
\end{tabular}

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.

\section*{Example 2:}

Study the two-dimensional pivot table below:
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{Two dimension pivot table - effect of total qualifier - \(\square^{\text {a }}\)} \\
\hline \multirow[t]{2}{*}{Grp} & Month & sum(Val) & above(sum(Val) & above(total sum(Val)) \\
\hline & 1 & 1 & & \\
\hline \multirow{3}{*}{A} & 2 & 3 & 1 & 1 \\
\hline & 3 & 5 & 3 & 3 \\
\hline & Total & 9 & & \\
\hline \multirow[t]{4}{*}{B} & 1 & 2 & & 5 \\
\hline & 2 & 4 & 2 & 2 \\
\hline & 3 & 6 & 4 & 4 \\
\hline & Total & 12 & & \\
\hline Total & & 21 & - & \\
\hline
\end{tabular}

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.

\section*{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.

\section*{Example 1:}

Study the two-dimensional pivot table below:
\begin{tabular}{|c|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{rowno() and noofrows()} & - \(\square\) \\
\hline Month & Grp & rowno() & rowno(total) & noofrows() & noofrows(total) \\
\hline \multirow[t]{3}{*}{1 ■} & A & 1 & 1 & 2 & 6 \\
\hline & B & 2 & 2 & 2 & 6 \\
\hline & Total & 0 & - & 2 & \\
\hline \multirow[t]{3}{*}{2} & A & 1 & 3 & 2 & 6 \\
\hline & B & 2 & 4 & 2 & 6 \\
\hline & Total & 0 & - & 2 & \\
\hline \multirow[t]{3}{*}{3 E} & A & 1 & 5 & 2 & 6 \\
\hline & B & 2 & 6 & 2 & 6 \\
\hline & Total & 0 & - & 2 & \\
\hline \multicolumn{2}{|l|}{Total} & - & 0 & - & 6 \\
\hline
\end{tabular}

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.

\section*{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 1 st level partial sums and entries not expanded in the 3rd column.
1 for 2 nd level partial sums and entries only expanded in the 1 st column.
0 for grand total cell.
dimensionality ()
\begin{tabular}{|c|c|c|c|}
\hline \multicolumn{4}{|l|}{Pivot table -} \\
\hline Country & Team & Salesman & dimensionality() \\
\hline \multirow[t]{3}{*}{Denmark \(\square\)} & \multirow[t]{2}{*}{Copenhagen \({ }^{\text {E }}\)} & Preben & 3 \\
\hline & & Total & 2 \\
\hline & \multicolumn{2}{|l|}{Total} & 1 \\
\hline \multicolumn{3}{|l|}{Germany \({ }^{\text {( }}\)} & 1 \\
\hline \multirow[t]{4}{*}{Sweden \(\quad\) -} & \multirow[t]{2}{*}{Gothenburg} & John & 3 \\
\hline & & Total & 2 \\
\hline & \multicolumn{2}{|l|}{Lund \({ }^{\text {a }}\)} & 2 \\
\hline & \multicolumn{2}{|l|}{Total} & 1 \\
\hline \multicolumn{3}{|l|}{Total} & 0 \\
\hline
\end{tabular}
\begin{tabular}{|l|l|l|c|}
\hline \multicolumn{3}{|l|}{ Straight table } \\
\hline Country & Team & Salesman & dimensionality() \\
\hline \multicolumn{3}{|l|}{} \\
\hline Denmark & Copenhagen & Preben & 3 \\
\hline Germany & Berlin & Helga & 3 \\
\hline Sweden & Gothenburg & John & 3 \\
\hline Sweden & Lund & Ann & 3 \\
\hline Sweden & Lund & Peter & 3 \\
\hline
\end{tabular}

\section*{secondarydimensionality ()}
\begin{tabular}{|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{Pivot table - -} \\
\hline Country 0 & \multicolumn{3}{|l|}{\(\square^{\text {D }}\) Denmark} & \(\pm\) Germany 0 \\
\hline Salesman & Total & \(\square\) Preb & & \\
\hline Team & & Total & Copenhagen & \\
\hline & & & & 1 \\
\hline
\end{tabular}

\section*{72 Nested Aggregations and Related Issues}

This chapter exemplifies some important techniques in relation to nested aggregations and the use of the Advanced Aggregation (page 289) function in charts.

\section*{Note!}

As of QlikView version 9 no more than 100 levels of nesting is allowed.

\section*{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.

\section*{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 289) function in combination with calculated dimensions, see Add calculated dimension... (page 602).

\section*{Example:}

The following data has been read from the script:
\begin{tabular}{|l|l|}
\hline \multicolumn{2}{|l|}{ Original data from script } \\
\hline SalesRep & Customer \\
\hline Donna Brown & Bechtel Corporation \\
\hline Karl Anderson & Berkeley Design \\
\hline Donna Brown & Capitolnet Marketing Group (CMG) \\
\hline Karl Anderson & Chas T. Main, Inc. \\
\hline Karl Anderson & Degolyer and MacNaughton \\
\hline Lisa Taylor & ediSys \\
\hline John Doe & Fimetrics Systems \\
\hline Kathy Clinton & HCS \\
\hline Lisa Taylor & Homestead Custom \\
\hline Lisa Taylor & Illuminati \\
\hline John Doe & Metro-Goldwyn-Mayer, Inc. \\
\hline Lisa Taylor & Onetouch Interactive \\
\hline Peggie Hurt & Savetz Publishing \\
\hline William Fisher & TECC \\
\hline William Fisher & VA Research \\
\hline Lisa Taylor & XYZ Operations \\
\hline
\end{tabular}

An obvious question given this data would be: "How many customers does each sales rep have?". This is easily done in a standard chart:
\begin{tabular}{|l|r|}
\hline \multicolumn{2}{|l|}{ First question: How many customer does each sales } \\
rep have?
\end{tabular} - \(\quad\) count(Customer) \begin{tabular}{|lr|}
\hline SalesRep & 2 \\
\hline Donna Brown & 2 \\
\hline John Doe & 3 \\
\hline Karl Anderson & 1 \\
\hline Kathy Clinton & 5 \\
\hline Lisa Taylor & 1 \\
\hline Peggie Hurt & 2 \\
\hline William Fisher & \\
\hline
\end{tabular}

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 289) 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:
\begin{tabular}{|l|r|}
\hline \begin{tabular}{l} 
2nd question: How many salesreps have 1, 2, 3 etc \\
customers?
\end{tabular} \\
\hline -aggr(count(Customer),SalesRep) & count(distinct SalesRep) \\
\hline 1 & 2 \\
\hline 2 & 3 \\
\hline 3 & 1 \\
\hline 5 & 1 \\
\hline
\end{tabular}

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 289) function may of course contain calculated dimensions, which in turn make use of the Advanced Aggregation (page 289) function. It should however be relatively easy to loose track of what you are doing when passing the third level of aggregation.

\section*{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.

\section*{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:
\begin{tabular}{|l|l|r|}
\hline \multicolumn{3}{|l|}{ Original data from script } \\
\hline Class & Name & - \\
\hline English & John & \\
\hline English & Karen & 5 \\
\hline English & Lisa & 1 \\
\hline History & John & 4 \\
\hline History & Karen & 3 \\
\hline History & Lisa & 3 \\
\hline Math & John & 2 \\
\hline Math & Karen & 3 \\
\hline Math & Lisa & 3 \\
\hline
\end{tabular}

We must now make a chart with Class as dimension and \(\boldsymbol{\operatorname { m a x }}\) (Score) as expression. A straight table with sum of rows will look like follows:
\begin{tabular}{|l|r|}
\hline \multicolumn{2}{|l|}{ Straight table with sum of rows } \\
\hline Class & - \\
\hline History & \(\max\) (Score) \\
\hline Math & 3 \\
\hline English & 4 \\
\hline & 5 \\
\hline
\end{tabular}

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:
\begin{tabular}{|l|r|}
\hline Pivot table with expression total & \multicolumn{1}{|c|}{} \\
\hline Class & \(\max (\) Score \()\) \\
\hline English & 5 \\
\hline History & 3 \\
\hline Math & 4 \\
\hline Total & 5 \\
\hline
\end{tabular}

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:
\begin{tabular}{|l|r|}
\hline \multicolumn{2}{|l|}{ Pivot table with sum of rows } \\
\hline Class & sum(aggr(max(Score),Class)) \\
\hline English & 5 \\
\hline History & 3 \\
\hline Math & 4 \\
\hline Total & 12 \\
\hline
\end{tabular}

As you see the total is again the one desired. What happened?
Well the beauty of the Advanced Aggregation (page 289) 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 289) functions.
For the total row however, the Advanced Aggregation (page 289) 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.

\section*{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.

\section*{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 \(\{, f l d\}\rangle]]\)-expression, \(x\)-expression \([, y 0[, x 0]])(\) page 776) and linest_b ([\{set_expression\}][ distinct ] [ total [<fld \{.fld \}>] ] y-expression, \(x\)-expression [, y0 [, x0 ]] ) (page 776) 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 289) function containing the same base expression and dimension(s) as the containing chart. The Advanced Aggregation (page 289) 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 converted to a straight table, where the regression values are shown in cells.
\begin{tabular}{|c|c|c|c|c|c|}
\hline \multicolumn{5}{|l|}{Trend line calculated as expression with linest aggregation} & - \(\square\) \\
\hline X & Y & linreg \(\mathrm{y}=\mathrm{mx}+\mathrm{b}\) & R2 & m & b \\
\hline 0 & 533 & 733,5 & 0,33628 & -118,0 & 733,5 \\
\hline 1 & 859 & 615,5 & 0,33628 & -118,0 & 733,5 \\
\hline 2 & 612 & 497.5 & 0,33628 & -118,0 & 733,5 \\
\hline 3 & 222 & 379,5 & 0,33628 & -118,0 & 733,5 \\
\hline
\end{tabular}

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)

```

\section*{73 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.

\subsection*{73.1 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 737)).

\subsection*{73.2 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.

\section*{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".

\subsection*{73.3 Calculated Formula}

The syntax for expression in calculated labels is virtually the same as for Chart Expressions (page 765), with a few exceptions:
- With no dimensions to iterate over, the Chart Aggregation functions (page 770) 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.

\section*{Example:}
\(=\) Currency
is equal to
```

= only(Currency)

```

\section*{74 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 600)).
Any fields can be grouped together.
Drill-down groups are marked with an icon,
while cyclic groups are displayed with an \(\mathcal{F}\) 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 drillup icon in the chart.

\subsection*{74.1 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:

\section*{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 169) 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 drillup 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.

\section*{Note!}

Although it is convenient to use this feature with natural hierarchies, there is nothing preventing its use in other situations as well.

\subsection*{74.2 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 609) )! Cyclic dimension groups are made up of a number of Fields (page 169), whereas the cyclic display in chart expressions cycles through a number of Chart Expressions (page 765).

\section*{75 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.

\section*{Standard Messages}

\section*{Custom Messages}

Apply to All

Clear All

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.

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... ).

Click this button to apply the custom messages to all the calculated objects of the document.

Click this button to clear all the custom error messages.

\section*{Part 7 Number Formats}

\section*{76 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.

\subsection*{76.1 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.

\subsection*{76.2 Number Interpretation}

\section*{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 354)) or the operating system settings (Control Panel).

\section*{Example:}


\section*{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 354).)
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 354)) 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.

\section*{Example:}


\subsection*{76.3 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 354)) or with the help of a formatting function, see Formatting Functions (page 356). Finally it is also possible to reformat the data in the properties sheet of the sheet object.

\section*{Examples:}
1997-08-06
is stored as
35648
\begin{tabular}{lll}
\(09: 00\) & is stored as & 0.375 \\
1997-08-06 09:00 & is stored as & 35648.375 \\
or vice versa & & \\
35648 & \begin{tabular}{l} 
with number format \\
\\
\\
\\
'D/M/YY' is shown as
\end{tabular} & \(6 / 8 / 97\) \\
& \begin{tabular}{l} 
with number format \\
'hh.mm' is shown as
\end{tabular} & 09.00
\end{tabular}

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.

\section*{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:
\begin{tabular}{|l|}
\hline 0.375 \\
\hline 33857 \\
\hline \(97-08-06\) \\
\hline 970806 \\
\hline \(8 / 6 / 97\) \\
\hline
\end{tabular}

Inside QlikView:
A1) QlikView interpretation (without the special interpretation function in the script)
String value (list box) Number value
\begin{tabular}{|r}
\hline \\
\hline \(\mathbf{0 . 3 7 5}\) \\
\hline \\
\hline \(\mathbf{9 7 - 0 8 - 0 6}\) \\
\hline \(\mathbf{8 / 6 / 9 7}\) \\
\hline
\end{tabular}
0.375

33857
35648
970806

A2) With number format date 'YYYY-MM-DD'
```

String value (list box)
Number value

```

1899-12-30
0.375

33857
35648
970806

A3) With number format date 'MM/DD/YYYY'


A4) With number format time 'hh:mm'
String value (list box)
\begin{tabular}{|cc|}
\hline & \(09: 00\) \\
\hline & \(00: 00\) \\
\hline & \(00: 00\) \\
\hline & \(00: 00\) \\
\hline \(8 / 6 / 97\) & \\
\hline
\end{tabular}

A5) With number format fix '\# \#\#0.00'


Number value
0.375

33857
35648 970806

B1) QlikView interpretation with the special interpretation function date\# ( \(A\), 'M/D/YY')


B2) With number format date 'YYYY-MM-DD'


\section*{Number value}


35648

B3) With number format date 'MM/DD/YYYY'


B4) With number format time 'hh:mm'


B5) With number format fix '\# \#\#0.00'


\section*{Number value}
```

35648

```

\section*{Number value}


Number value


\section*{77 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.

\subsection*{77.1 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".

\section*{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. 14 fa ). 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.

\subsection*{77.2 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.

\subsection*{77.3 Times}
- To describe the hours, use the symbol "h" for each digit.
- To describe the minutes, use the symbol " m " for each digit.
- To describe the seconds, use the symbol "s" for each digit.
- To describe the fractions of a second, use the symbol "f" for each digit.
- To describe the time in AM/PM format, use the symbol "tt" after the time.
- Arbitrary separators can be used.

\subsection*{77.4 Time Stamps}

The same notation as that of dates and times above is used.

\section*{Examples of Format Codes}

\section*{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.

\section*{Examples (Special Number Formats):}
\begin{tabular}{lll} 
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)') & & 5 J
\end{tabular}
\begin{tabular}{lll} 
num(199, '(rom)') & returns & cxcix \\
num(199, '(ROM)' ) & returns & CXCIX
\end{tabular}

\section*{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.
WWWW YY-MM-DD describes the date as Saturday 01-03-31.

\section*{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

\section*{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

\section*{Part 8 Macros and Automation}

\section*{78 Using Automation and Macros with QlikView}

\subsection*{78.1 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.

\subsection*{78.2 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 \(\mathrm{C}++\) 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.

\section*{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:

\section*{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.

\section*{Sheet events:}
5. A macro can be run after a sheet is activated.
6. A macro can be run when a sheet is deactivated.

\section*{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.

\section*{Button events:}
9. A button sheet object can be linked to a macro.

\section*{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.

\section*{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.

\section*{79 Internal Macro Interpreter}

QlikView objects are accessible via automation from inside QlikView by means of the built-in scripting engine.

\subsection*{79.1 Invoking Macros}

Macros written in VBScript or JScript, inside a QlikView document can be invoked in a number of different ways:

\section*{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 425) page.

\section*{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 452) page.

\section*{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 452) page.

\section*{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 532) page.

\section*{Field Events:}
- A macro can be run when a selection has been made in a specified field, see Fields (page 169).
- 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 425) page.

\section*{Variable Events:}
- A macro can be run when a new value is directly entered into a specified Variable (page 171).
- 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 425) page.

\section*{Macro Editor:}

Irrespective of how they are triggered, macros are created and edited in the Edit Module (page 875) dialog.

\section*{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

```

\subsection*{79.2 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 qvilib 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

```

\subsection*{79.3 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 82) 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.
\begin{tabular}{ll} 
(List of entry points) & \begin{tabular}{l} 
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.
\end{tabular} \\
(Message) & \begin{tabular}{l} 
The current state of any error messages is displayed in the box above the \\
macro edit box.
\end{tabular} \\
Check & \begin{tabular}{l} 
Once you have a written sub, check and validate its syntax by pressing this \\
button.
\end{tabular} \\
Goto & \begin{tabular}{l} 
Highlighting an entry point from the List of entry points and clicking Goto, \\
takes you directly to its position in the Macro Edit Box.
\end{tabular} \\
Test & \begin{tabular}{l} 
With an entry point selected, you may test a macro by clicking the Test but- \\
ton. Any errors will appear in the Message box.
\end{tabular} \\
Debug & \begin{tabular}{l} 
Begin by selecting one of the subroutines to be debugged. By clicking the \\
Debug button, the debug mode is entered.
\end{tabular} \\
Scripting Engine & \begin{tabular}{l} 
Select between VBScript and JScript.
\end{tabular} \\
Requested Module Secu- \\
rity & \begin{tabular}{l} 
By selecting Safe Mode 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 Safe Mode, the execution will fail.
\end{tabular} \\
By selecting System Access mode the document designer ensures that the end
\end{tabular}

\section*{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. A very useful QlikView file describing the use of the macro functionality in QlikView - APIguide.qvw - is included with the general examples that are installed with the program.

\subsection*{79.4 Macro Debugger}


\section*{The Macro Debug page of the Edit Module dialog}

Before you can use the macro debugger, you must install the Microsoft Script Debugger.
By pressing the Debug button in the Edit Module dialog you will enter macro debug mode. In this mode you may step through the macro and inspect variable contents as you go along.
Before debug mode is entered you must select a macro (subroutines without parameters only) from the list and set one or more breakpoints in the code. Breakpoints can be toggled at any time by clicking on a line number next to a position in the code or by positioning the cursor in the line and pressing F9. Pressing CTRL+SHIFT+F9 will clear all breakpoints.
When entering debug mode two new panes appear in the dialog window. In the lower left you will find a pane indicating the current call stack. By double-clicking in the list the editor window will be scrolled to the position of the call clicked upon. The execution pointer will not be moved however.
In the lower right you will find a pane with current variables. By double-clicking one of the variables you will open the Variables dialog where the properties of the variable can be further examined and its value changed interactively.
In debug mode a separate menu becomes available. Many of the items on this menu have a keyboard shortcut counterpart. The keyboard shortcuts can be individually customized in the Editor page of User Preferences. The shortcuts mentioned below are the installation defaults.

Continue Continues execution of the macro to the next breakpoint or the end of the macro. Equivalent to pressing f5.

Step Into Steps execution one line. If the next line is a call to a subroutine execution will be stepped to the first line of the subroutine. Equivalent to pressing 88.
\begin{tabular}{ll} 
Step Out & \begin{tabular}{l} 
Executes the remaining statements of the current subroutine. Equivalent to \\
ctrl+shift+f8.
\end{tabular} \\
Step Next & \begin{tabular}{l} 
Steps execution one line. If the next line is a call to a subroutine the sub- \\
routine will be executed in its entirety and stopped at the first line after the \\
subroutine call. Equivalent to pressing shift \(+\mathrm{f8}\).
\end{tabular} \\
Variable Window & \begin{tabular}{l} 
Opens a window where the values of different variables can be inspected and \\
changed.
\end{tabular} \\
Expression Window & \begin{tabular}{l} 
Opens a dialog where you may enter and evaluate a VBScript or JScript expres- \\
sion based on current variable values.
\end{tabular} \\
Clear Breakpoints & \begin{tabular}{l} 
Clears the breakpoints set in the code for debugging. Equivalent to pressing \\
ctrl+shift+f9.
\end{tabular} \\
Stop Debugging & \begin{tabular}{l} 
Exits debug mode and returns to the normal Edit Module dialog.
\end{tabular}
\end{tabular}

\section*{80 Getting Hold of a QlikView Document}

\subsection*{80.1 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.

\section*{Example (Does not work in VBScript!):}
```

Private Sub OpenAndReload_Click()
Set QvDoc = GetObject
("c:\windows\desktop\test.qvw")
QvDoc.Reload
End Sub

```

\subsection*{80.2 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.

\section*{Examples:}
```

Sub Clr
Set QVDoc = ActiveDocument
QvDoc.ClearAll(false)
End sub
Sub EndQV
ActiveDocument.Application.Quit()
End sub

```

\section*{81 VBScript Function Calls from Script}

VBScript or JScript functions defined in the macro module of a QlikView document can be called from the script. If a function called is not recognized as a standard script function, a check will be made to see if a custom functions resides in the module. This gives you a large degree of freedom to define your own functions. Using custom macro functions will of course be somewhat slower than executing the standard functions.
```

rem *************************************************
rem ***THIS IS VBSCRIPT CODE FROM THE MODULE*********
rem***************************************************
rem ********* Global variables *********************
dim flag
rem *******functions accessible from script **********
rem ************* wrap for input box *********************
function VBin(prompt)
VBin=inputbox (prompt)
end function
rem ************ clear global flag **********************
function VBclearFlag()
flag=0
end function
rem ********* test if reference has passed ************
function VBrelPos(Ref, Current)
if Ref=Current then
VBRelPos="Reference"
flag=1
elseif flag=0 then
VBRelPos="Before "\&Ref\&" in table"
else
VBRelPos="After "\&Ref\&" in table"
end if
end function
// ***************************************************
// *************THIS IS THE SCRIPT
// **************************************************
let MaxPop=VBin('Max population in millions :');
// Ask limit
let RefCountry=VBin('Reference country :');
// Ask ref.

```
```

let dummy=VBclearFlag(); // Clears the global flag

```
Load
Country,recno(),
Capital,
"Area(km.sq)",
"Population(mio)",
VBrelPos('\$(RefCountry)', Country)
as RelativePos
from countryl.csv
(ansi, txt, delimiter is ',', embedded labels)
where "Population(mio)" <= \$(MaxPop);

\subsection*{81.1 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.

\section*{82 Using Macros in QV Documents on the QV-Server}

\subsection*{82.1 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.

\section*{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.

\subsection*{82.2 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

\section*{Note!}

Layout operations acting on the properties of Server objects are supported.

\subsection*{82.3 Macro trigger limitations}

The following triggers will work as usual in the QlikView Server environment regardless of which client is used:
```

Document.OnAnySelect
Field.OnSelect
Field.OnChange
Field.OnUnlock
Document.OnPressMacroButton
Variable.OnChange
Variable.OnInput

```

The following triggers are not to be used in the QlikView Server environment, since they lack meaning or may cause unexpected results:

OnPostReduceData
OnPostReload

\subsection*{82.4 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.

\subsection*{82.5 Server Side Export}

It is possible to export the content of any chart to a text file on the server by using a Macro connected to a Button. This feature has to be added in the QlikView document and then the macro can be executed from the QlikView Server directly in the web browser.
In the QlikView document QWT.qvw provided on the CD, you find a sheet called "Server Side Export". As the button is clicked, the content of the Straight Table CH20 can be exported as a text file to the server directory for QlikView Server. The exported data can then easily be read into Excel or other programs.
Below you find a sample of VB Script code used in the Edit Module in QlikView for making the export possible. The macro is then connected to a trigger, in the case of this example, a button.
```

Sub ServerExportEx
Set st = ActiveDocument.GetSheetObject("CH20")
st.ServerSideExportEx "C:\ProgramFiles\QlikView Server\CH20export.skv"
, ";" ,0
End Sub

```

\section*{Part 9 Appendix}

\section*{A Limitations and Requirements}

\section*{A. 1 Limitations of the Amount of Data to be Loaded}

The amount of data that can be loaded into a QlikView document is very large. It is primarily limited by the amount of primary memory of the computer. However, there are some inherent limitations in QlikView that one must be aware of when building very large documents.
A QlikView document cannot have more than:
\begin{tabular}{ll} 
Number of fields & only limited by RAM \\
Number of tables & only limited by RAM \\
Number of distinct values in one field & only limited by RAM
\end{tabular}

Each table, loaded through one script statement, or concatenated through several script statements, cannot have more than:

Number of cells only limited by RAM
Number of rows only limited by RAM

\section*{A. 2 System Requirements}

This version of QlikView is designed to run on an IBM PC compatible computer running Windows \({ }^{\mathrm{TM}}\). This section describes the hardware and software requirements to run QlikView on your computer.

\section*{Hardware}

\section*{You need a computer with}
- Intel Core Duo or higher recommended. QlikView makes full use of modern multi-core processor architecture and we thus strongly recommend using such processors.
- Graphics hardware supporting color XGA resolution, or better.
- A monitor with a resolution of at least \(1024 \times 768\).
- A mouse or an equivalent pointing device supported by MS Windows.
- A CD-ROM drive (if installation is to be made from CD).
- A hard disk with at least 150 MB of free disk space.
- Enough primary memory (see below).

\section*{Operating System}

One of the following operating systems is needed:
- Microsoft \({ }^{\circledR}\) Windows XP \({ }^{\text {TM }}\) SP3 or
- Microsoft \({ }^{\circledR}\) Windows Vista \({ }^{\mathrm{TM}}\) or
- Microsoft \({ }^{\circledR}\) Windows \(7^{\text {TM }}\) or
- Microsoft \({ }^{\circledR}\) Windows Server \(2003^{\text {TM }}\) or
- Microsoft \({ }^{\circledR}\) Windows Server 2008 \({ }^{\text {tM }}\)

\section*{Additional Requirements for QlikView x64}
- Processor supporting the x 64 architecture
- Intel Core 2 Duo or higher recommended
- 4 GB RAM
- A hard disk with at least 350 MB of free disk space to install
- Windows XP Professional x64 Edition \({ }^{\text {TM }}\) SP3 or
- Windows Vista x64 Edition \({ }^{\text {TM }}\) or
- Windows \(7 \times 64\) Edition \({ }^{\text {TM }}\) or
- Windows Server \(2003 \times 64\) Edition \({ }^{\text {TM }}\) or
- Windows Server \(2008 \times 64\) Edition \({ }^{\text {TM }}\) or
- Windows Server 2008 R2

\section*{Memory}

\section*{Primary Memory}

Although it is possible to run QlikView with less memory, the use of a computer with at least 1 GB for 32bit Windows and 4 GB for 64 -bit Windows is recommended.

The amount of memory you need for QlikView depends on the amount of data you load, and on the data structure. Too little primary memory will drastically affect the performance of QlikView in a negative way. If you notice that your computer accesses the hard disk every time you make a selection in QlikView, you either have too many programs running, or too little primary memory.

\section*{Virtual Memory}

Make sure that your swap file (the virtual memory) is large enough. The swap file should in principle be as large as possible. However, if you have limited disk space, or a lot of primary memory, you may want a smaller swap file or no swap file.
The system usually takes care of the swap file size dynamically. However, if you have too little free space on the hard disk, you may get problems with your swap file getting too small. In such a case, you can fix the size of the swap file. This is done in the Virtual memory dialog, which is accessed from the task bar: choose Settings, Control Panel, System, Performance, Virtual Memory.

\section*{Note!}

32-bit Windows sets a limit of 2 GB logical address space for any one program running (3 GB for Windows Advanced Server). QlikView can never make use of more memory than this, regardless of the size of physical memory in the computer. This limitation does not exist when running QlikView 64-bit edition.

\footnotetext{
Note!
Defragment the hard disk before changing the swap file settings. It usually leads to a larger contiguous free hard disk area and thus a larger possible swap file.
}

\section*{A. 3 Dynamic Link Libraries (dll's) Needed}

In the windows system directories of all the clients there must be a number of dll's. They are all installed during the installation procedure. For information on the dll's required, see the ReadMe file in your QlikView directory. QlikView uses Windows Installer and msi technology for installation, which ensures that the correct versions of system dll's are always used.

\section*{B Keyboard Command Shortcuts}

In this chapter you will find a list of the different keyboard command shortcuts available in QlikView．

\section*{B． 1 File Menu Command Shortcuts}
\begin{tabular}{|c|c|c|}
\hline CTRL＋N & Equivalent to New on the File menu． & \(\square\) \\
\hline CTRL＋O & Equivalent to Open．．．on the File menu． & 3 \\
\hline CTRL＋SHIFT＋O & Equivalent to Open in Server．．．on the File menu． & \\
\hline CTRL＋S & Equivalent to Save on the File menu． & \(\square\) \\
\hline F12 & Equivalent to Save As．．．on the File menu． & \\
\hline CTRL＋P & Equivalent to Print．．．on the File menu． & 象 \\
\hline CTRL＋SHIFT + P & Equivalent to Print As PDF．．．on the File menu． & 嗗 \\
\hline CTRL＋E & Equivalent to Edit Script．．．on the File menu． & 3 \\
\hline CTRL＋R & Equivalent to Reload on the File menu． & 怱 \\
\hline CTRL＋SHIFT＋R & Equivalent to Partial Reload on the File menu． & \\
\hline CTRL＋T & Equivalent to Table Viewer．．．on the File menu． & 时 \\
\hline \multicolumn{3}{|l|}{B． 2 Edit Menu Command Shortcuts} \\
\hline CTRL＋Z & Equivalent to Undo Layout Change on the Edit menu． & 4 \\
\hline CTRL＋Y & Equivalent to Redo Layout Change on the Edit menu． & C \\
\hline CTRL＋X & Equivalent to Cut on the Edit menu． & \％ \\
\hline CTRL＋C & Equivalent to Copy on the Edit menu． & 成 \\
\hline CTRL＋V & Equivalent to Paste on the Edit menu． & 遇 \\
\hline DEL & Equivalent to Remove on the Edit menu． & \\
\hline CTRL＋A & Equivalent to Activate All on the Edit menu． & \\
\hline CTRL＋F & Equivalent to Search on the Edit menu． & 鲑 \\
\hline CTRL + SHIFT + F & Equivalent to Advanced Search on the Edit menu． & \\
\hline
\end{tabular}

\section*{B. 3 View Menu Command Shortcuts}
\begin{tabular}{ll} 
CTRL+Q & Equivalent to Current Selections... on the View menu. \\
CTRL+G & Toggles layout design grid on or off.
\end{tabular}

\section*{B. 4 Selections Menu Commands Shortcuts}
\begin{tabular}{ll} 
SHIFT+LEFTARROW & Equivalent to Back on the Selections menu. \\
SHIFT+RIGHTARROW & Equivalent to Forward on the Selections menu. \\
CTRL+SHIFT+L & Equivalent to Lock on the Selections menu (locks all selections). \\
CTRL+SHIFT+U & Equivalent to Unlock on the Selections menu (unlocks all selections). \\
CTRL+SHIFT+D & Equivalent to Clear on the Selections menu (clears all selections).
\end{tabular}

\section*{B. 5 Settings Menu Command Shortcuts}
\begin{tabular}{ll} 
CTRL+ALT+U & Equivalent to User Preferences... on the Settings menu. \\
\(\mathbf{C T R L}+\mathbf{A L T}+\mathbf{D}\) & Equivalent to Document Properties... on the Settings menu. \\
CTRL+ALT+S & Equivalent to Sheet Properties... on the Settings menu. \\
\(\mathbf{C T R L}+\mathbf{A L T}+\mathbf{V}\) & Equivalent to Variable Overview... on the Settings menu. \\
\(\mathbf{C T R L + A L T + E ~}\) & Equivalent to Expression Overview... on the Settings menu
\end{tabular}

\section*{B. 6 Bookmarks Menu Command Shortcuts}
\begin{tabular}{lll} 
CTRL+B & Equivalent to Add Bookmark... on the Bookmarks menu. \\
CTRL+SHIFT+B & Equivalent to More... on the Bookmarks menu.
\end{tabular}

\section*{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.

\section*{B. 8 Object Menu Command Shortcuts (List box, Statistics box and Open Multi Box)}
\begin{tabular}{ll} 
CTRL+L & Equivalent to Lock on the Object menu (locks selections in active object). \\
CTRL+U & \begin{tabular}{l} 
Equivalent to Unlock on the Object menu (unlocks selections in active \\
object).
\end{tabular} \\
CTRL+D & Equivalent to Clear on the Object menu (clears selections in active object). \\
ALT+ENTER & \begin{tabular}{l} 
Equivalent to Properties... on the Object menu (opens the Properties dialog \\
of an active object).
\end{tabular}
\end{tabular}

\section*{B. 9 Script Keyboard Shortcuts}

To see a list of the available keyboard shortcuts in the Script type ctrl+qse in the script pane.
\begin{tabular}{|c|c|}
\hline CTRL+G & Go to line number in the script. \\
\hline CTRL+K,C & Comment lines in script. \\
\hline CTRL+K, \({ }^{\text {d }}\) & Uncomment lines in script. \\
\hline CTRL+Q,T,A & Add tab in script. \\
\hline CTRL+Q,T, P & Promote active tab. \\
\hline CTRL+Q,T,D & Demote active tab. \\
\hline CTRL+Q,T,N & Rename active tab. \\
\hline CTRL+Q,T,R & Remove active tab. \\
\hline CTRL+Q,Q & Creates an autogenerated script. \\
\hline CTRL + Q,U,I & Opens Unicode input utility. \\
\hline CTRL+Q,J,P,G & Creates a script for reading attributes from jpeg files. \\
\hline CTRL+Q,M,P,3 & Creates a script for reading attributes from mp3 files. \\
\hline CTRL + Q,W,M,A & Creates a script for reading attributes from wma files. \\
\hline
\end{tabular}

\section*{B. 10 F Key Keyboard Shortcuts}
F1 Activates context sensitive help.

F3
F6
F7
F12
CTRL+F6
CTRL+F7

Activates context sensitive help.
Enters search mode if a searchable object is activated.
Activates the immediate left sheet tab of the currently active tab.
Activates the immediate right sheet tab of the currently active tab.
Equivalent to Save As... on the File menu.
Activates the leftmost sheet tab.
Activates the rightmost sheet tab.

\section*{C Application Performance Optimization}

\section*{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.

\section*{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.

\section*{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.

\section*{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 ;

```

\section*{Note!}

The aggregation is done over Alfa as this is the dimension in the test.

\section*{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.

\section*{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 379)).

\section*{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.

\section*{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 0 in it.

\section*{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.

\section*{D Frequently Asked Questions}

\section*{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
www.appdeploy.com
www.wise.com

\section*{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 512 MB 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.

\section*{D. 3 Scripts and Loading Data}

Q: Is it possible to use more than one Binary statement in a QlikView script?
A: No unfortunately not. Binary load uses very special logic to read the data already processed once into QlikView in another QlikView document. The "unpacking" used requires a "clean slate" in memory which is the reason why a Binary statement must be the very first statement in the script. Therefore it is also impossible to use Binary more than once in a script. However you can use QVD files to consolidate data from multiple QlikView documents at the same speed as with Binary.

Q: Is the only requirement for an automatic join that the field names are the same?
A: Yes! Such a join is called an association.

Q: Can I achieve an automatic join between two fields with different field names?
A: Yes, but you must rename one of the fields in the script using an ALIAS statement or an AS clause.

Q: Can I use Where clauses, Group by clauses and other SQL-like syntax on text files?
A: QlikView offers you a rich script language for text files including large parts of SQL syntax and a number of additional features.

Q: Can I use Where clauses, Group By clauses and other SQL-like syntax on binary QlikView files?
A: No.

Q: What is the difference between an association between 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 196) and File Wizard: Type - Fixed Record (page 198).

Q: Can I update a QlikView document by loading only the data that have changed?
A: Yes, see QVD Files (page 393) 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 200)) 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.

\section*{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 \(\mathrm{n} / \mathrm{a}\) ?
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 proceeded by a distinct qualifier.

Q: Can I mix AND and OR logic for value selections within a list box?
A: No, the QlikView logic prohibits this.

Q: Can I have OR logic between list boxes?
A: Only indirectly. An advanced alternative is to use semantic links to transfer selections made in a list box to another field. 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 \(\operatorname{text}()\) 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.

\section*{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.

\section*{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 261) and Access Restriction Table Wizard (page 193)) 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 401).

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 192)) .

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.

\section*{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.

\section*{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.

\section*{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 backups of the data files.

\section*{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.

\section*{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.

\section*{F Backus-Naur Formalism}

The QlikView command line syntax (Command Line Syntax (page 38)) 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.
\(::=\quad\) 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.

\section*{Example:}

The description of the alias statement is:

\section*{aliasfieldnameasaliasname \{, 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 fieldnameasaliasname 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};

```

\section*{Glossary}

\section*{A}

\section*{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.

\section*{access restriction}

Method to restrict access of a QlikView document for different users of user groups.

\section*{active}

A window, dialog box, tabbed sheet or sheet object that is currently in use is called active or current.

\section*{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( \(\operatorname{Aggr}(\operatorname{Count}(\ldots), \ldots))\).

\section*{aggregation function}

An aggregation function is a function that operates on a set of values - several records to return a single scalar value. Examples: \(\operatorname{Sum}(), \operatorname{Count}(), \operatorname{Avg}()\). 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.

\section*{AJAX}

Shorthand for "Asynchronous JavaScript and XML", is a development technique for creating interactive web applications. See also QlikView AJAX zero footprint client.

\section*{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.

\section*{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.

\section*{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.

\section*{ASCII}

American Standard Code for Information Interchange. A standardized set of characters for computers.

B
biff
The native Excel file format.

\section*{blue}

Color of a cell that is locked.

\section*{bookmark}

A saved set of selections which can be recalled by a user and shared with other users.

\section*{bookmark object}

A sheet object where bookmarks can be created, deleted and recalled.

\section*{C}

\section*{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.

\section*{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.

\section*{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 circlular reference needs to be broken. QlikView does this by setting one of the tables as loosely coupled.

\section*{clear selections}

An operation that resets a QlikView application to a state where no selections are made, i.e. no field values are excluded.

\section*{column}

Term commonly used instead of field in databases.

\section*{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.

\section*{concatenate (2)}

An operation that uses two text strings and combines them into one. The operator used for this is the ampersand "\&".

\section*{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.

\section*{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.

\section*{crosstable}

A QlikView script prefix used to load cross tables.

\section*{CSV}

CSV is a file extension for comma separated value files. Used when a table is stored as a text file.

\section*{current selections box}

A sheet object that shows selections in the fields and their logical status.

\section*{custom object}

A placeholder sheet object for custom OCX controls in QlikView controls.

D

\section*{Data source}

Data source is a name for a physical database, defined through the ODBC interface.

\section*{DBMS}

DBMS (DataBase Management System). The database program used in the storage and modification of data in the database.

\section*{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'.

\section*{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.

\section*{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.

\section*{document}

A QlikView file saved in binary format having the default extension qvw. Also called QlikView file.

\section*{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.

\section*{dual fields}

Fields that have both textual and numeric representations, e.g. dates, months, formatted numbers, etc.

\section*{E}

\section*{edit script dialog}

The text editor where the load script is edited.

\section*{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.

\section*{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.

\section*{F}

\section*{field}

Equivalent to column in database. In QlikView, a field is typically represented by a list box (see also value and list box).

\section*{font}

The typesetting for letters and characters. Fonts are described by name, appearance and size, as in "Arial bold 10 pt ".

\section*{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.

\section*{format painter}

The paint brush on the tool bar. It can be used to transfer properties from one sheet object to an other.

\section*{formatting}

Dual fields need formatting and often this needs to be explicitly done using the format( ) function.

\section*{G}

\section*{generic}

A QlikView script prefix used to unpack and load generic databases.

\section*{gray}

Color of a cell that is excluded.

\section*{green}

Color of a cell that has been selected.

\section*{H}

\section*{help, html help}

The normal Windows help that is invoked from the Help menu or by hitting the F1 key.

\section*{hidden script}

A part of the load script that can be hidden and password protected.

I

\section*{in-memory analysis}

Term for BI tools that hold the entire data set in primary memory and calculate all necessary aggregations on demand.

\section*{input box}

A sheet object that is used for entering data into QlikView variables and displaying their values.

\section*{input field}

A field that is declared as an Input field can be used for input. Good for planning, forecasting and budgeting.

\section*{intervalmatch}

A QlikView script prefix used to match discrete values to intervals.

\section*{J}

\section*{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.

\section*{K}

\section*{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.

\section*{key field}

A field that exists in two tables and connects the two.

L

\section*{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.

\section*{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).

\section*{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.

\section*{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.

\section*{\(\log\) file}

A file that, like a log book, logs everything that happens during a script execution.

\section*{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.

\section*{M}

\section*{mapping}

A QlikView script prefix that enables the script developer to use lookup tables.

\section*{module}

A dialog where your VB Script macros can be entered, tested and stored.

\section*{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.

\section*{N}

\section*{nested aggregation}

A calculation using a two-step aggregation. See aggr function.

\section*{0}

\section*{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.

\section*{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.

\section*{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.

\section*{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

\section*{P}

\section*{path}

The location of a file, folder or directory on a disk. See absolute/relative path.

\section*{pivot table}

A sheet object that allows several dimensions, several expressions with aggregated data, pivoting and grouping.

\section*{primary memory}

The memory that the computer uses for programs that are running. Usually in the form of RAM.

\section*{Q}

\section*{QlikView (.qvw) file}

A QlikView (.qvw) file saved in binary format has the default extension qvw. Also called document.

\section*{QlikView AccessPoint}

Single point of access to corporate QlikView documents. Requires QlikView Server.

\section*{QlikView AJAX zero footprint client}

A lightweight object-based client to the QlikView Server that is based on the AJAX technology.

\section*{QlikView Desktop}

QlikView with a bought user license, can be used to create documents and also to open documents created by other users.

\section*{QlikView OCX}

QlikView packaged as an ActiveX component for integration into third party software.
Not to be confounded with OCX replacement controls.

\section*{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.

\section*{QlikView Plug-In}

Short for QlikView Desktop for Microsoft Internet Explorer.

\section*{QlikView Publisher}

Administrative tool for automated reload and publishing of QlikView documents. QlikView Publisher is an add-on module to QlikView Server and its capabilities are part of the standard QlikView Server installation package.

\section*{QlikView Server}

Software for publishing QlikView documents to online users using a client-server architecture.

\section*{QlikX}

Object-based QlikView OCX for integration into other products or into web sites.

\section*{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.

\section*{R}

\section*{RAM}

Abbreviation of Random Access Memory. Usually synonymous to primary memory.

\section*{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.

\section*{record}

Equivalent to a row in a table.

\section*{red}

Color of a cell that has been explicitly excluded by the user.

\section*{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.

\section*{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.

\section*{report}

With "report" one usually refers to a static paper or pdf report.

\section*{report editor}

QlikView has a built-in report editor in which one can prepare standard paper reports for printing.

\section*{reserved fields}

Fields used in the access restriction management: USERID, PASSWORD, SERIAL ACCESS, NTNAME, NTDOMAINSID, NTSID and OMIT.

\section*{resident}

A table that has been loaded in the script can be accessed using a Load ... resident statement in the script.

\section*{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.

\section*{S}

\section*{scalar function}

A scalar function is a function that operates on a single value to return a single scalar value. Examples: \(\operatorname{chr}(), \sin ()\), applymap( ). See also Aggregation function and Range function.

\section*{Script}

A description of what data to load. The script is a small program that is run by QlikView. When executed, it connects your QlikView document to one or several data sources or opens text files and reads the specified information into QlikView. The data source and the fields to be included are defined here.

\section*{Search Object}

Search objects can be used for searching for information anywhere in the document.

\section*{Section Access}

A section of the QlikView load script that can be used for access restriction.

\section*{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.

\section*{semantic}

A QlikView script prefix used to load semantic links. These are used for linking one selection to an other, e.g. self-references within a field.

\section*{separator}

A character or code that separates one column or field from the next. Sometimes referred to as delimiter.

\section*{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.

\section*{Slider/calendar object}

A sheet object which can be used to select values in a field or set values to one or two variables.

\section*{SQL}

SQL, Structured Query Language, is a standard for making queries in relational databases.

\section*{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.

\section*{statement}

A script command can also be called statement. All statements must end with semicolons ";".

\section*{statistics box}

A sheet object that displays a set of statistical entities, calculated on the possible field values of the corresponding field.

\section*{stored procedures}

Programs that are stored in, and run on Databases.

\section*{straight table}

A sheet object that allows several dimensions as well as several expressions with aggregated data, and has good sorting capabilities.

\section*{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.

\section*{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.

\section*{system variables}

System variables are parameters generated by QlikView in the script for special purposes, e.g. definitions of document number formats

\section*{T}

\section*{tab in script}

The script can be split up onto several tabs. This improves the structure and facilitates navigation in the script.

\section*{tabbed sheet}

See sheet.

\section*{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.

\section*{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.

\section*{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.

\section*{U}

\section*{URL}

Universal Resource Locator. A general address of the type used on the World Wide Web.

\section*{V}

\section*{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 alphanumeric (text) data. See dual fields.

\section*{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.

\section*{W}

\section*{wizard}

A series of dialogs to help the user achieve a given goal, e.g. creating a chart.

\section*{X}

X64
Computer architecture for 64-bit technology.

\section*{XML}

Extended Markup Language. More complex than html, but not as complex as SGML. One XML file can contain one or several tables.



\begin{tabular}{|c|c|c|c|}
\hline data from files & 168 & Document Properties: Layout & 438, 476 \\
\hline Data Source & 196 & Document Properties: Number & 434 \\
\hline Data sources & 168 & Document Properties: Opening & 414 \\
\hline Data Viewer & 181 & Document Properties: Presentation & 433 \\
\hline database & 168 & Document Properties: Scheduler & 420 \\
\hline date & 357 & Document Properties: Security & 423 \\
\hline Date and time functions & 338 & Document Properties: Sheets & 416 \\
\hline date\# & 827 & Document Properties: Sort & 432 \\
\hline dateformat & 277, 355 & Document Properties: Tables & 429 \\
\hline day & 339 & Document Properties: Triggers & 425 \\
\hline DayEnd & 352 & Document Properties: Variables & 421, 524 \\
\hline DaylightSaving & 342 & Document Reports & 135 \\
\hline DayName & 352 & Document Support Information & 65 \\
\hline DayNames & 278, 356 & Dollar-sign expansion & 279 \\
\hline DayStart & 351 & Drag and drop to Microsoft Office & 456 \\
\hline Debugger & 216 & Drop Field & 231 \\
\hline decimalsep & 276, 354 & Drop Table & 232 \\
\hline Define expression & 737 & dual & 359 \\
\hline Definition & 611 & \multicolumn{2}{|l|}{E} \\
\hline Delimiter & 251 & & \\
\hline Design Toolbar & 58 & e & 318 \\
\hline Dimension & 600 & Edit expression & 737 \\
\hline Dimension Limits & \[
\begin{aligned}
& 55,663,675, \\
& 695,709,730
\end{aligned}
\] & Edit menu
Edit Module & 57
875 \\
\hline dimensionality & 815, 843 & Edit script & 68 \\
\hline Dimensions \(640,647,65\) & \[
\begin{aligned}
& 75,683,689 \\
& 709,721,730
\end{aligned}
\] & Edit Script dialog box Embedded image files & 173
165 \\
\hline Directory & 230 & embedded labels & 252 \\
\hline disconnect & 230 & Environmental Functions & 330 \\
\hline distinct & 246, 262 & Error variables & 278 \\
\hline distinctrow & 262 & ErrorMode & 278 \\
\hline div & 303, 805 & evaluate & 323 \\
\hline Do..loop & 231 & even & 307, 808 \\
\hline Document data source & 196 & Examples of Aggregate Qualifiers & 831 \\
\hline Document Functions & 337 & Examples of Chart Inter Record functions & 839 \\
\hline Document Properties & 408 & Examples of Chart Ranking functions & 835 \\
\hline Scrambling & 436 & Execute & 232 \\
\hline Server & 418 & exists & 326 \\
\hline Document Properties: Caption & 442, 479 & Exit script & 233 \\
\hline Document Properties: Font & 88, 437, 475 & exp & 312 \\
\hline Document Properties: General & 408 & explicit labels & 252 \\
\hline Document Properties: Groups & 427 & Exponential and logarithmic functions & 312 \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|c|}
\hline getcurrentfield & 330, 824 & InDay & 345 \\
\hline getcurrentselections & 330, 824 & InDayToTime & 346 \\
\hline getexcludedcount & 331, 825 & Independent Samples T-test & 763 \\
\hline getfieldselections & 331, 824 & index (string function) & 320-321 \\
\hline getfolderpath & 335 & Indicator & 109 \\
\hline getnotselectedcount & 331,825 & Info & 239 \\
\hline getpossiblecount & 331, 825 & inline & 247 \\
\hline GetRegistryString & 331 & Inline Data Wizard & 192 \\
\hline getselectedcount & 331, 824 & InLunarWeek & 345 \\
\hline Getting Started Wizard & 45 & InLunarWeekToDate & 345 \\
\hline GMT & 342 & InMonth & 343 \\
\hline Green & 362 & InMonths & 344 \\
\hline Grid Chart & 673 & InMonthsToDate & 344 \\
\hline Grid Chart: Object Menu & 673 & InMonthToDate & 344 \\
\hline group by & 248, 262 & Inner & 239 \\
\hline Group Settings & 428 & Input & 332 \\
\hline H & & Input Box & 519 \\
\hline & & Input Box Properties & \\
\hline Has Error Bars & 609 & Presentation & 523 \\
\hline hash128 & 324 & Input Box Properties: Constraints & 421, 524 \\
\hline Hash160 & 324 & put Box Properties: General & 521 \\
\hline Hash256 & 324 & Input Box Properties: Number & 526 \\
\hline having & 262 & Input Box: Object Menu & 519 \\
\hline Header size & 252 & Input field & 241 \\
\hline Help menu & 64 & Input field aggregation function & 797 \\
\hline Hidden script & 68, 181 & Input Fields & 170 \\
\hline hideprefix & 274 & inputavg & 798 \\
\hline hidesuffix & 274 & Inputfield & 241 \\
\hline Hierarchic groups (drill-down) & 853 & inputsum & 798 \\
\hline Hierarchy & 236 & InQuarter & 343 \\
\hline Hierarchy Parameters & 214 & InQuarterToDate & 343 \\
\hline HierarchyBelongsTo & 237 & Insert and Update & 397 \\
\hline hour & 338 & Insert only & 396 \\
\hline HRank & 823 & Insert Update and Delete & 398 \\
\hline HSL & 361 & Inter record functions & 326 \\
\hline I & & Internal Files & 165 \\
\hline if & 328 & Internal Macro Interpreter & 873 \\
\hline If..then..elseif..else..end if & 238 & Internal VBScript Interpreter & 873 \\
\hline Image stretch & 542 & Interpretation functions & 825 \\
\hline Image_size & 239 & interval & 359 \\
\hline Import Bookmark & 129 & interval\# & 829 \\
\hline include & 274 & IntervalMatch & 241, 243 \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|c|}
\hline ltrim & 322 & MonthEnd & 348 \\
\hline LunarweekEnd & 351 & MonthName & 348 \\
\hline LunarWeekName & 351 & MonthNames & 277, 355 \\
\hline LunarWeekStart & 350 & MonthsEnd & 349 \\
\hline M & & MonthsName & 349 \\
\hline & & MonthsStart & 348 \\
\hline mac & 250, 391 & MonthStart & 348 \\
\hline Macro & 279, 452, 873 & Moving selection & 109 \\
\hline Macros 4 & 532, 537, 873 & MsgBox & 332 \\
\hline Macros on QlikView Server & 883 & msq & 252 \\
\hline Magenta & 362 & Multi Box & 489 \\
\hline makedate & 340 & Multi Box Properties: General & 491 \\
\hline maketime & 341 & Multi Box Properties: Number & 498 \\
\hline makeweekdate & 340 & & 4 \\
\hline Map ... using & 254 & rites. Presentat & 9 \\
\hline Mapping & 253 &  & 9 \\
\hline Mapping functions & 325 & \multirow[t]{2}{*}{My Server Objects} & \multirow{3}{*}{579} \\
\hline MapSubstring & 325 & & \\
\hline match & 328 & N & \\
\hline \multicolumn{3}{|l|}{Mathematical constants and parameter-free functions318avigation Toolbar} & 58 \\
\hline max & 287, 770-771 & Nested aggregations & 845 \\
\hline MaxString & 288, 770, 773 & networkdays & 352 \\
\hline median & 291, 781 & New Chart & 45, 590, 597, 637 \\
\hline Mekko chart & 707 & New file & 67 \\
\hline Mekko Chart: Object Menu & 707 & New Variable & 421, 524 \\
\hline mid & 321 & no eof & 251 \\
\hline min & 287, 770-771 & no labels & 252 \\
\hline Mini chart settings & 468 & no quotes & 252 \\
\hline Minstring & 770 & NoConcatenate & 254 \\
\hline MinString & 288, 772 & Non-hierarchic groups (cyclic) & 854 \\
\hline minute & 338 & NoOfColumns & 821 \\
\hline missingCount & 770 & NoOfFields & 337 \\
\hline MissingCount & 289, 774 & NoOfFields() & 337 \\
\hline mixmatch & 329 & NoOfReports & 337 \\
\hline mod & 304, 805 & NoOfRows & 337, 821, 842 \\
\hline mode & 287, 770, 772 & NoOfRows() & 337 \\
\hline money & 357 & NoOfTables & 337 \\
\hline money\# & 827 & NoOfTables() & 337 \\
\hline moneydecimalsep & 277, 355 & normdist & 314 \\
\hline MoneyFormat & 277, 355 & norminv & 314 \\
\hline moneythousandsep & 276, 354 & not & 284, 768 \\
\hline month & 339 & now & 340 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|c|}
\hline nPer & 316 & Outer & 255 \\
\hline NPV & 294, 784 & \multirow[t]{2}{*}{P} & \\
\hline null & 330 & & \\
\hline Null functions & 330 & Page Settings & 144 \\
\hline NullAsNull & 255 & Paired T-test & 762 \\
\hline NullAsValue & 255 & peek & 327 \\
\hline  & & permut & 307, 808 \\
\hline nullcount & 770 & Personal Edition & 29 \\
\hline NullCount & 289, 774 & & \\
\hline nulldisplay & 275 & pi & 18 \\
\hline nullinterpret & 275 & pick & 328 \\
\hline nullvalue & 276 & Pie Chart & 681 \\
\hline num & 35 & Pie Chart: Object Menu & 681 \\
\hline num\# & 826 & Pivot Table & 715 \\
\hline num\# & 826
306,807 & Pivot Table: Object Menu & 718 \\
\hline \multirow[t]{2}{*}{Number \(474,644,651,659,6\)} & 306,807 & Pmt & 316 \\
\hline & \[
\begin{aligned}
& 680,685,691, \\
& , 713,725,736
\end{aligned}
\] & Pop-up Window Settings & 411 \\
\hline Number Format & 434 & pow & 313 \\
\hline Number interpretation variables & 276, 354 & Predefined values & 422, 525 \\
\hline Numbers on Data Points & 609 & previous & 326 \\
\hline numcount & 305, 807 & Print & 67 \\
\hline Numeric operators & 284,768 & Print as PDF & 68 \\
\hline numericcount & 770 & Print as PDF (Object) & 68 \\
\hline NumericCount & 289, 773 & Print Options: Date \& Time & 103 \\
\hline nummax & 306, 807 & Print Options: General & 99 \\
\hline nummin & 306, 807 & Print Preview & 104 \\
\hline numsum & 305, 806 & Print: General & 99 \\
\hline \multirow[b]{2}{*}{0} & & Print: Header/Footer & 101 \\
\hline & & Print: Layout & 100 \\
\hline Object Caption Style & 410 & Project Files & 51 \\
\hline Object menu & 63, 456 & Properties files & 51 \\
\hline odd & 307, 808 & purgechar & 323 \\
\hline oem & 250, 391 & PV & 317 \\
\hline only & 287, 770, 772 & \multicolumn{2}{|l|}{Q} \\
\hline Open Database Files & 190 & \multirow[b]{2}{*}{QlikTechBlue} & \\
\hline Open file & 67 & & 363 \\
\hline Open Internet Files & 191 & QlikTechGray & 363 \\
\hline Operators & 283, 767 & QlikView Desktop & 29 \\
\hline OR & 284,768 & QlikView file & 51 \\
\hline ord & 320 & QlikView Personal Edition & 29 \\
\hline order by & 248, 262 & QlikViewProject Files & 51 \\
\hline osuser & 330 & qlikviewversion & 332 \\
\hline othersymbol & 276 & Qualify & 256 \\
\hline & & QuarterEnd & 347 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|c|}
\hline QuarterName & 347 & RangeSum & 308, 809 \\
\hline QuarterStart & 347 & RangeTextCount & 310, 812 \\
\hline Quick Chart Wizard & 747 & RangeXIRR & 312 \\
\hline Quotation marks & 281 & RangeXNPV & 312 \\
\hline Quotes & 252 & rank & 822 \\
\hline QVD files & 393 & Rate & 317 \\
\hline QVD files and Incremental Load & 394 & recno & 318 \\
\hline QvdCreateTime & 336 & Record length & 252 \\
\hline QvdFieldName & 336 & Red & 362 \\
\hline QvdNoOfFields & 336 & Redo Layout & 68 \\
\hline QvdNoOfRecords & 336 & Reexecute script & 68 \\
\hline QvdTableName & 336 & Reference Lines & 626, 667, 679 \\
\hline QVPath & 274 & Refresh & 67 \\
\hline QVRoot & 274 & Relational operators & 285, 768 \\
\hline quuser & 330 & Reload & 68 \\
\hline QVWorkPath & 274 & reloadtime & 330 \\
\hline QVWorkRoot & 275 & Rem & 257 \\
\hline R & & Rename Field & 258 \\
\hline & & Rename Table & 258 \\
\hline Radar Chart & 653 & Renaming fields & 369 \\
\hline Radar Chart: Object Menu & 653 & repeat & 322 \\
\hline rand & 318 & Replace & 260, 324 \\
\hline Range functions (Chart) & 809 & Report & 134 \\
\hline Range functions (Script) & 308 & Report Editor & 134 \\
\hline RangeAvg & 308, 809 & Report Settings & 141 \\
\hline RangeCorrel & 311, 815 & ReportComment & 337 \\
\hline RangeCount & 308, 810 & ReportID & 337 \\
\hline RangeFractile & 309, 811 & ReportName & 337 \\
\hline RangeIRR & 311 & ReportNumber & 337 \\
\hline RangeKurtosis & 309, 811 & Reports & 133 \\
\hline RangeMax & 309, 810 & Reports menu & 62, 133 \\
\hline RangeMaxString & 310, 813 & Reset Clear State & 69 \\
\hline RangeMin & 308, 810 & resident & 248 \\
\hline RangeMinString & 310, 813 & RGB & 361 \\
\hline RangeMissingCount & 310, 813 & Right & 259, 321 \\
\hline RangeMode & 310, 814 & round & 305, 806 \\
\hline RangeNPV & 311 & rowno & 319, 820, 842 \\
\hline RangeNullCount & 310, 812 & rtrim & 31,820,842 322 \\
\hline RangeNumericCount & 309, 812 & & \\
\hline RangeOnly & 311, 814 & & \\
\hline RangeSkew & 309, 811 & Sample & 261 \\
\hline RangeStdev & 309, 811 & Scatter Chart & 661 \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|c|}
\hline Statistics Box: Object Menu & 483 & TableNumber() & 337 \\
\hline Statistics Chart Wizard & 761 & Tag & 269 \\
\hline Status bar & 75 & \(\tan\) & 313 \\
\hline stdev & 286, 290, 781 & tanh & 313 \\
\hline stdv & 290, 774 & tdist & 314 \\
\hline sterr & 291, 775 & text & 829 \\
\hline steyx & 291, 775 & Text Object & 539 \\
\hline Store & 267 & Text Object Properties: General & 540 \\
\hline Straight Table & 727 & Text Object: Object Menu & 539 \\
\hline Straight Table: Object Menu & 727 & Text on Data Point & 609 \\
\hline String functions & 320 & TextBetween & 323 \\
\hline String operators & 284, 768 & textcount & 770 \\
\hline StripComments & 275 & TextCount & 289, 774 \\
\hline Style \(\quad 510,640,648,656,665\) & \[
\begin{aligned}
& 676,684,690, \\
& 709,724,735
\end{aligned}
\] & \begin{tabular}{l}
Theme Maker Wizard \\
Themes
\end{tabular} & \(583-584\)
581 \\
\hline Sub..end sub & 267 & thousandsep & 276, 354 \\
\hline subfield & 322 & time & 358 \\
\hline substringcount & 324 & Time Chart Wizard & 755 \\
\hline sum & 286, 770-771 & time\# & 828 \\
\hline Switch..case..default..end switch & 268 & timeformat & 277, 355 \\
\hline Synthetic dimension functions & 803 & timestamp & 360 \\
\hline SysColor & 361, 364 & timestamp\# & 828 \\
\hline System fields & 365 & timestampformat & 277, 355 \\
\hline System Functions & 330 & TimeZone & 342 \\
\hline System variables & 274 & tinv & 315 \\
\hline T & & Title Settings & 596 \\
\hline Table Box & 501 & today & 340 \\
\hline Table Box Properties: General & 505 & Toolbars & 58 \\
\hline Table Box Properties: Number & 511 & Tools menu & 62 \\
\hline Table Box Properties: Presentation & 508 & top & 817, 839 \\
\hline Table Box Properties: Sort & 507 & Trace & 270 \\
\hline Table Box: Object Menu & 501 & Transform Table & 201 \\
\hline Table file & 168 & Tree View in list boxes & 460, 481 \\
\hline Table File Wizard: Type & 196 & Trellis & 605 \\
\hline Table format & 251 & Trigonometric and hyperbolic functions & 313 \\
\hline Table functions & 336 & trim & 322 \\
\hline Table labels & 273 & true & 318 \\
\hline Table Viewer & 181 & TTest_conf & 296, 788 \\
\hline TableName & 337 & TTest_df & 295, 786 \\
\hline TableName() & 337 & TTest_dif & 296, 787 \\
\hline \multirow[t]{2}{*}{TableNumber} & 337 & TTest_lower & 296, 788 \\
\hline & & TTest_sig & 296, 787 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|c|}
\hline TTest_sterr & 296, 787 & User Preferences: Folders & 94 \\
\hline TTest_t & 295, 786 & User Preferences: General & 77 \\
\hline TTest_upper & 296, 788 & User Preferences: Mail & 93 \\
\hline TTest1_conf & 299, 792 & User Preferences: Objects & 86 \\
\hline TTest1_df & 298, 791 & User Preferences: Print & 92 \\
\hline TTest1_dif & 299, 791 & User Preferences: Registration & 97 \\
\hline TTest 1_lower & 299, 792 & User Preferences: Save & 81 \\
\hline TTest1_sig & 298, 791 & User Preferences: Security & 96 \\
\hline TTest1_sterr & 299, 792 & User Reports & 135 \\
\hline TTest1_t & 298, 791 & Using the Pivot Table & 715 \\
\hline TTest1_upper & 299, 792 & UTC & 342 \\
\hline TTest1w_conf & 300, 794 & UTF-8 & 250, 391 \\
\hline TTest1w_df & 300, 793 & \multicolumn{2}{|l|}{V} \\
\hline TTestlw_dif & 300, 793 & \multirow[b]{2}{*}{Value handling variables} & \\
\hline TTest1w_lower & 300, 794 & & 275 \\
\hline TTestlw_sig & 300, 793 & Variable & 171 \\
\hline TTest1w_sterr & 300, 794 & Variable Overview & 161 \\
\hline TTest1w_t & 299, 793 & Verbatim & 275 \\
\hline TTest1w_upper & 301, 795 & View menu & 58 \\
\hline TTestw_conf & 298, 790 & VRank & 824 \\
\hline TTestw_df & 297, 789 & \multicolumn{2}{|l|}{W} \\
\hline TTestw_dif & 297, 789 & Waterfall chart & 610 \\
\hline TTestw_lower & 298, 790 & week & 339 \\
\hline TTestw_sig & 297, 789 & weekday & 339 \\
\hline TTestw_sterr & 297, 790 & WeekEnd & 350 \\
\hline TTestw_t & 297, 788 & WeekName & 350 \\
\hline TTestw_upper & 298, 790 & WeekStart & 349 \\
\hline \multicolumn{2}{|l|}{U} & weekyear & 339 \\
\hline Undo Layout & 68 & when & 272 \\
\hline Unicode & 250, 391 & where & 248, 262 \\
\hline Unless & 270 & Where Clause & 212 \\
\hline Unlock & 111 & while & 248 \\
\hline Unlock and Clear All & 69 & White & 362 \\
\hline Unlock Selections & 69 & WildMatch & 329 \\
\hline Unmap & 271 & Window menu & 63 \\
\hline Unqualify & 271 & WinPath & 275 \\
\hline Untag field & 271 & WinRoot & 275 \\
\hline upper & 322 & X & \\
\hline User Preferences Editor & 82 & XIRR & 294, 783 \\
\hline User Preferences: Design & 85 & XML & 51 \\
\hline User Preferences: Export & 89 & xml files & 253 \\
\hline
\end{tabular}
\begin{tabular}{lrl} 
XNPV & & 294,784 \\
xor & & 284,768
\end{tabular}
year 339
year2date 341
YearEnd 346
YearName 347
YearStart 346
Yellow 363
Z
\begin{tabular}{ll} 
ZTest_conf & 301,796 \\
ZTest_dif & 301,795 \\
ZTest_sig & 301,795 \\
ZTest_sterr & 301,796 \\
ZTest_z & 301,795 \\
ZTestw_conf & 302,797 \\
ZTestw_dif & 302,797 \\
ZTestw_sig & 302,796 \\
ZTestw_sterr & 302,797 \\
ZTestw_z & 302,796
\end{tabular}```


[^0]:    Clear List The button clears the list of Recent Servers on the Start Page.

[^1]:    Note!
    The QlikView versions of the source and target computers should be the same in order to guarantee $100 \%$ functionality.

[^2]:    Print, Header/Footer

[^3]:    Report List>>
    Takes you back to the List of Reports page.

[^4]:    The second page of the Alert wizard

[^5]:    The Table Viewer dialog

